NO STAMP.

APPLICATION OF PSL TO CONSTRUCTION PROCESS INFORMATION SPECIFICATION AND EXCHANGE

Tesfagaber, G.

Ph.D. Thesis 2004

Department of Civil & Building Engineering

Loughborough University

Abstract

This study explores the potential use of PSL as a standard language for process information exchange between software applications in construction The applicability of PSL in construction was investigated, and the impact of the language on software interoperability was evaluated

To investigate the applicability of PSL in construction, the construction process information representations (data types) within the pre-construction stages were identified through a technical analysis of the supporting software applications. Then the construction process information representations' (the applications data types) semantic concepts were mapped against the manufacturing process concepts of the PSL ontology to determine the level of applicability of the PSL language in providing neutral definitions for the construction process information representations and establish the need for new extensions to PSL ontology to incorporate construction process concepts

To evaluate the impact of the PSL language on construction software interoperability, the language was implemented in a construction scenario within the pre-construction stages for process information exchange between AutoCAD, CCS, and Microsoft project software applications.

A number of conclusions were drawn from the results of the research, PSL is applicable in construction provided that extensions are developed to the PSL ontology to incorporate some of the construction process concepts that the current PSL language can not capture; PSL is a promising standard language for process information exchange between heterogeneous construction software applications, the use of PSL in construction as a standard language offers the potential for integration within construction and between sectors such as construction and manufacturing.

Finally recommendations are drawn for further research to provide information on the issues that need further investigation and methods of research

Acknowledgments

I would like to express my gratitude for the precious and countless support and encouragement of my supervisor, prof. Dino Bouchlaghem This project could not have been finished without the support and encouragement of prof Dino Bouchlaghem, who has been a great support during my study, has persistently supported me in structuring the project and writing, and helped me in resolving problems that could have affected my study.

Additionally I would like to express my thankfulness to.

- Prof. Chimay Anumba, a director of the project for his support and encouragements
- Prof Anne-Francoise Cutting-Decelle, University of Evry, a visiting Fellow to Loughborough University, who made information about the ISO standards and developments available for my study
- The UK Construction Computer Software, CCS software support groups in South Africa, Glasgow and Dorset, particularly, Mr Dave corfield at Glasgow CCS office who has made the software licence available during my study and provided me with useful technical information about the CCS software
- Mr W.D Sher, who was a quantity surveying lecturer at the Department of Civil and building Loughborough University, has introduced me to CCS basics and provided me with further and useful information about CCS

Table of Contents

	Page
Abstract	i
Acknowledgments	ii
Table of Contents	iii
List of Figures	xi
List of Tables	xiv
List of Abbreviations	xv

CHAPTER 1 Introduction	1
1 0 Introductıon	1
1.1 Background	1
1.1.1 Constructıon Projects and Processes	1
1.1 2 IT Supports for Constructıon Processes	2
1.1 2.1 Current Construction Informatıon Technology (CIT) Practıces	4
1.1 2.2 Informatıon in the Pre-Constructıon Process Stages	6
1.1.2.3 The Need for an Agreed Interpretatıon of Process Models	9
1.2 Problem Definitıons and Research Goal	10
1.2 1 Aim and Objectıves of the Research Study	12
1.3 The Research	13
1.4 Justıficatıon for the Research Study	15
1.5 Research Issue and Context of the Project	17
1.6 Structure of the Thesıs	18
1.7 Delımıtatıon of the Scope	20
CHAPTER 2 Construction Information Modelling and Integration	21
2.1 What ıs Construction Informatıon?	21
2.2 Informatıon Categonsation and Organisation	22
2.2.1 Subject of Informatıon	22
2.2.2 View, Aspects and Perspectıves of Informatıon	23
2.3 Objectıves of Modellıng Construction Informatıon	23
2.4 Issues ın Modelling Constructıon Information	24
2.5 Approaches to Informatıon Modellıng	25

2 5 1 Product Modelling Approach 25

2.5.1 1 Background and Objectives of Product Modelling 25

2 5 1.2 Concepts of Product Modelling 26

2 5 2 Process Modelling Approach 28

2 5 2 1 Backgrounds and Objectives of Process Modelling 28

2 5 2 2 Purpose of Process Modelling 29

2 5 2 3 Concepts of Process Modelling 30

2.6 Modelling Methodologies and Techniques 31

2 7 IDEF Family of Modelling Methodologies Overview 33

2.8 Classification of Process Modelling Methodologies 35

2 8 1 General Systems Modelling (GSM) 36

2.8.1 1 Structured Analysis and Design Technique (SADT) 36

2 8.1.2 Strength and Weakness of SADT 37

2.8.2 IDEF0 Functional Modelling Methodology 37

2 8 2.1 Functional Modelling Overview 37

2 8.2.2 IDEF0 Modelling Approach 40

2 8.2.3 Basic Concepts and Syntax of IDEF0 41

2.8.2 4 The IDEF0 Functional Model Objective 43

2.8.2.5 Strengths and Weaknesses of IDEF0 44

2.8.3 IDEF3 "Process Flow and Object State Description Capture" Method 44

2.8.3.1 Overview 44

2.8.3 2 Process Descriptions and Models 45

2 8.3.3 Basic Concepts of IDEF3 Technique 46

2.8.3.4 IDEF3 Representation of Process Descriptions 46

2 8.3.5 Scenarios: The Organizing Structure for IDEF3 Process Descriptions 47

2.8.3.6 Description Representation Concepts and Syntax of IDEF3 Schematics 47

2.8.3.7 Strengths and Weaknesses of IDEF3 50

2.8.4 Knowledge Representations 50

2.8.4.1 IDEF5 Ontology Description Capture Method 51

2.8.4.2 Knowledge Interchange Format (KIF) 54

2.8 5 Behavioural Modelling 63

2.8.5.1 Petri Nets 63

2.8.6 Data Flow Diagrams (DFD) 64

2.9 Summary and Conclusions 65

CHAPTER 3 Information Standardisation for Construction 68

3.1 Introduction 68

3 2 Information Standardisation Definitions and Objectives 69

3 2 1 Neutral Interchange Format 70

3 3 Information Representation Standardisation for AEC/FM: State-of-the-Art 72

3.3.1 The ISO 10303 STEP Standard Product Data Oriented Representation 72

3.3.1 1 The STEP Model Development Methodology 74

3 3 1 2 Main Features of the ISO 10303 STEP Standard 75

3.3.1 3 Components of ISO 10303 STEP Standard 76

3 3 2 The IAI's IFC De Facto Standard. Product Data Oriented Representation 80

3.3 2.1 The IFC Model Architecture 81

3 3 2.2 The IFCs Architectural Principles 84

3 3 2 3 The IFCs Architectural Model Schemata 85

3 3 2.4 Connecting External Models to the IFC Model 85

3.3.3 Classification Standard. (ISO TC59 SC13 TR 14177: ISO/PAS 12006) 86

3 3.3 1 Conceptual Schema of Construction Works 89

3.3 4 Other Standardization Developments 92

3.3 5 Standard Types Available in the Construction Industry 93

3 4 Summary and Conclusions 97

CHAPTER 4 Information Standardisation for Manufacturing **100**

4.1 Process Specification Language (PSL) 100

4 2 Scope of PSL 101

4.3 Goal of PSL 101

4 4 PSL Related Works 102

4 5 Development of PSL 104

4 5 1 PSL Implementations in Manufacturing Scenarios 106

4 5 2 PSL Extensions Development Approach 107

4.6 Semantics of Process Representations 107

4.7 The State of PSL: PSL ontology 108

4.7.1 The Language 109

4.7.2 Model Theory 110

4.7.3 Proof Theory 110

4.8 Consistency and Completeness of the PSL Ontology 112

4.9 PSL Architecture 112

3.9.1 Informal Semantics of PSL-Core 113

4.10 Requirements for PSL Extensions 114

4.10.1 Non-logical Lexicon 115

4.10.2 Specification of Models 115

4.10.3 Axioms of the Extensions 115

4.10.4 Grammar for Process Descriptions 115

4.10.5 Format for Extensions 116

4.11 ISO 18629 PSL Standards 116

4.11.1 Organization of the ISO 18629 PSL Standard 117

4.12 ISO 18629-1x Series Core theories 118

4.12.1 ISO 18629-11 PSL-Core 118

4.12.2 ISO 18629-12 Outer Core 118

4.12.3 ISO 18628-13 Time and Ordering Theories 120

4.12.4 ISO 18629-14 Resource Theories 121

4.12.5 ISO 18629-15 Activity Performance Theories 121

4.13 ISO 18629-2x Series External Mappings 122

4.14 ISO 18629-4x Series Definitional Extensions 122

4.14.1 ISO 18629-41: Activity Extension 122

4.14.2 ISO 18629-42: Temporal and State Extension 123

4.14.3 ISO 18629-43: Activity Ordering and Duration Extension 125

4.14.4 ISO 18629-44 Resource Roles 126

4.14.5 ISO 18629-45 Resource Sets 126

4.14.6 ISO 18629-46 Processor Activities 127

4.14.7 ISO 18629-47 Process Intent 127

4.15 ISO 18629-2xx Series Translator Implementation Guidelines 127

4.16 Summary 128

CHAPTER 5 Research Methodology and Design 130

5.1 Literature Review and Theoretical Framework 130

5.1.1 Research Approaches in Construction IT 130

5.1.2 Philosophical Basis of Research 132

5.1.3 Critical Research 135

5.1.4 Triangulation or Pluralistic Research Approaches 135

5.1 5 Basic and Applied Research 136

5 2 Research Methods 137

5.2.1 Research Methods Available 137

5 2.1 1 Action Research 137

5 2 1 2 Case Study 138

5 2.1 3 Ethnographic Research 139

5 2 1 4 Grounded Theory 139

5 2.1.5 Concept Mapping 140

5 3 Modes of Analysis 143

5 3 1 Approaches to Data Analysis 143

5 4 Valid Analysis 145

5 4 1. Criteria for Evaluating Research 145

5.5 Research Strategy and Methods of the Study 148

5.5.1 Theoretical Framework 148

5 6 Research Methods Used 156

5 7 Data Collection and Analysis Methods Used 158

5.8 Conclusions 161

CHAPTER 6 Investigation of Applicability of PSL in Construction 162

6 1 Introduction 162

6.2 Construction Environment 163

6 2.1 Introduction 163

6 2.2 Construction Process Information 164

6 2 3 Construction Computer applications 165

6 2.4 Communication 166

6 2 5 The challenges 167

6 3 PSL from Construction Processes Perspective 167

6 3 1 Process Specification Requirements in the Pre-Construction Stages 168

6 3.1.1 Process Specification Requirements related Architectural Design 169

6.3.1.2 Process Specification Requirements related Cost Estimating 171

6 3 1.3 Process Specification Requirements related Scheduling and Planning 173

6 4 Construction Process Concepts Analysis and Mapping to PSL ontology 174

6.4.1 Introduction 174

6.4.2 PSL Ontology Concepts and Current Theories and Extensions 175

6 4 2 1 Core Theones of PSL Ontology 176

6 4 2 2 Definıtıonal Extensıons of PSL 178

6 4 3 Process Informatıon Concepts wıthın the Pre-Constructıon Stages 181

6 5 Analysıs of the Constructıon Process Representatıons and Mappıng to PSL Ontology 185

6 6 Conclusıon on the Applıcabılıty of PSL ın Constructıon 192

CHAPTER 7 A Scenarıo for Trial Implementation of PSL in Construction 194

7 1 Introductıon 194

7.2 Implementatıon Overvıew 195

7 2.1 Purpose of the Implementatıon of PSL ın a Constructıon Envıronment 195

7 2.2 Implementatıon Approach 196

7.3 Construction Context Overvıew 196

7.4 Detaıled Descnptıon of the Constructıon Scenarıo 199

7 4 1 Process Informatıon Representatıons Overvıew 199

7 4 1.1 Archıtectural Desıgn 200

7 4 1 2 Pre-Tender Cost Estımatıng 201

7 4 1.3 Schedulıng and Planning 202

7 4 2 Software Applıcatıons Supportıng Process Informatıon Representatıons 204

7 4 2.1 AutoCAD 204

7.4 2 2 Constructıon Computer Software (CCS) 206

7.4 2 3 Mıcrosoft Project Scheduler 208

7.4 3 Constructıon Projects Realization Processes Overvıew 210

7.4.4 An Example of a Construction Sıtuatıon wıthin the Scenarıo 211

CHAPTER 8 Implementation of PSL in the Construction Scenario 213

8.1 Introductıon 213

8 2 Cases Study: “Metal Door ın a Metal Openıng of a Buıldıng” Project 214

8.2.1 Products and Materials Overvıew of the Case Study Project 218

8 3 Detaıled View of the Case Study Project Construction Processes 218

8 3.1 Install Metal Door ın Metal Buıldıng Openıng 221

8 3.2 Construct Metal Buıldıng Openıng 221

8 3.3 Install Girt or Attach Top Gırt to Wood Frame 222

8 3 4 Install Wall Panel 223

8 3 5 Attach Door Assembly to Gırt and Wood Frame of the Buıldıng Openıng 224

8 3 6 Construct Door Frame 225

8.3 7 Prepare Doorȷambs 226

8 3 8 Construct and Hang Door Leaf 226

8 3 9 Place Hardware and Components and Fınısh Door 229

8 4 Process Informatıon Representatıons usıng Scenarıo Applıcatıons 230

8 4 1 AutoCAD Desıgn Data Representatıon usıng Excel Fıle Format 231

8 4.2 Cost Estımatıng Data Representatıon usıng CCS Fıles 234

8 4.3 Constructıon Plan Representatıon usıng Mıcrosoft Proȷect Fıles 235

8 5 Informatıon Translatıon between Scenarıo Software Applıcatıons 238

8 5 1 Semantıc Translatıon Defınıtıons for the Scenarıo Applıcatıons 239

8 5 2 Archıtectural desıgn data ın AutoCAD Applıcatıon's Representatıons 241

8.5.3 Archıtectural Design Data in PSL Syntax (KIF Notatıon) 242

8.5 4 Archıtectural Desıgn Data ın PSL Representatıons 245

8.5.5 Archıtectural Desıgn Data in PSL Syntax and CCS Termınology 245

8.5 6 Archıtectural Desıgn Data in CCS Applıcatıon's Input Fıle 251

8 5 7 Cost Estimating Output Fıle ın CCS Applicatıon's Representatıons 253

8 5 8 Cost Estımating Data in PSL Syntax and CCS Termınology 254

8 5 9 CCS Cost Estımatıng Data ın PSL Representations 256

8 5.10 Cost-Estımating Data ın PSL Syntax and MS Proȷect Termınology 258

8.5.11 Cost-Estımatıng Data ın MS Project Applıcatıon's Representations 260

8 5.12 MS Proȷect Schedulıng Data ın PSL Syntax & MS Project Termınology 269

8.5.13 MS Project Scheduling Data ın PSL Representatıons 271

CHAPTER 9 Discussion 274

9.1 Introductıon 274

9.2 Investıgatıon of the Applicabılıty of PSL ın Constructıon 274

9 3 Implementatıon of PSL ın the Pre-Constructıon Process Stages 281

9.3.1 The Constructıon Scenarıo 282

9 3.2 Capabılıty of the Scenarıo Applications to Exchange Informatıon 284

9 3.3 IDEF3 Modellıng of the Case Study Proȷect Constructıon Processes 285

9.3.4 Software Applıcations' Representatıon of the Process Informatıon 287

9.3.5 Informatıon Exchange between the Scenario Applıcatıons usıng PSL 287

9 3.6 Data Translatıon between Scenarıo Applicatıons' Representatıons 291

9 3 6 1 Translation Process 293

9 3 7 Proposal for New Extensions to PSL Ontology 295

9 4 Web Services As a Possible Alternative to the use of PSL 297

CHAPTER 10 Conclusions and Recommendations 300

10.1 Introduction 300

10 2 Conclusions 300

10 3 Recommendations for Further Research 302

References 305

Publications & other Reports 323

Appendices 325

A: Glossary of Terms for AutoCAD, CCS & Microsoft Project Applications 326

B: IDEF3 Model of Design, Cost Estimating & Scheduling Processes 332

C: Elaboration Forms for the IDEF3 Models 353

D: Ontology for the IDEF3 Models 444

E. Drawing Representation of the Case Study Project 529

List of Figures

Page

1.1 IFC 1.5 schema definition of the products/processes relationships between 7

2 1 The product modelling concepts (Eastman, 1992) view-models, aspects, sub model and a core model 28

2.2 Classifications of process modelling methodologies and techniques 36

2.3 IDEF0 activity model with ICOM notation 42

2.4 IDEF0 activity model decomposition overview 43

3 1 Alternative mechanisms for direct electronic information exchange 69

3 2 Three different models of knowledge sharing 71

3.3. The organisation of STEP standard 73

3.4 Example of an Application Activity Model (AAM) in IDEF0 notation 74

3.5 Overview of the STEP document architecture 77

3 6 STEP data specification 79

3 7 Layering concepts of IFC architecture (IFC TG, 2000) 82

3 8. Core extensions from kernel classes (IFC TG, 2000) 84

3 9 Basic concepts for describing construction works 90

3.10. Level order and main properties of construction works 91

3.11. Portions of an information model for AEC projects 93

4.1. A sampling of existing process representations 106

4 2. The need for semantics of terminologies 108

4 3. The PSL semantic architecture 113

4 4. Relationships between sets of axioms within PSL Outer Core 120

4 5. Definitional extensions of ISO 18629-41 123

5.1. Conceptual mapping of the software applications' data types to PSL ontology 150

5.2. PSL ontology component parts of interest to this project 153

5.3 Scenario 1 before and 2 after the implementation of PSL in the scenario 156

5.4. Summary of the research strategy 158

6.1. Design application model based on AutoCAD 170

6 2. Cost estimating application model based on CCS 172

6 3. Scheduling application model based on Microsoft Project planner 173

6.4. PSL ontology (manufacturing process concepts map) 181

6 5 AutoCAD design data types map 182

6 6 (a) CCS Input cost estımatıng data types map 183

6 6 (b) CCS Output cost estımatıng data types map 184

6 7 Mıcrosoft Project schedulıng data types map 185

6.8. Desıgn and cost estımatıng data types mappıng to PSL ontology 188

6 9 CCS Cost data types mappıng to PSL ontology 189

6 10 Mıcrosoft Project schedulıng data types to PSL ontology 190

7.1 Outlıne of constructıon scenarıo wıthın the pre-constructıon stages 199

7 2 IDEF3 model for the produce and manage archıtectural desıgn data hıgh level process 200

7 3 IDEF3 model for pre-tender procedure cost estımatıng hıgh level process 202

7 4 IDEF3 model for the pannıng & schedulıng hıgh level process 203

8 1. Hıerarchıcal breakdowns of the products and materıals for/of the case study project. 218

8.2. Generıc overvıew of buıldıng constructıon processes 219

8 3 Detaıled vıew of the "metal door in a metal openıng of a buıldıng project" constructıon process 220

8.4 "Install metal door ın metal buıldıng opening" process 221

8.5 Construct metal buıldıng openıng process 222

8 6. Install gırt or attach top gırt to wood frame process 223

8.7. Install wall panel process 224

8 8. Attach door assembly to gırt and wood frame of the buıldıng openıng process 225

8.9. Construct door frame process 226

8.10. Prepare doorjambs process 227

8.11 Construct and hang door leaf process 228

8.12. Place hardware and components and finish door process 230

8.13. Desıgn data of the case study project structured ın Excel file format 233

8.14. (a) Cost estimating data of the case study project structured ın CCS BoQs and Macro files 234

8.14. (b) Cost estımatıng data of the case study project structured ın CCS operatıon & work sheet files 235

8.15. (a) Gantt chart representation of the tasks and related ınformatıon for the case study project construction plan 236

8 15 (b) Resource Sheets representation of the resources and related information for the case study project construction plan 237

8 16 Process information exchange between AutoCAD, CCS & Microsoft project applications using PSL as a common/standard language 240

9 1. Flow of semantic information between scenario applications and the role of PSL 292

List of Tables

Page

2 1 Constsructıon ınformatıon at dıfferent levels of abstractıon 22

2 2 Suıtes of IDEF modellıng methodologıes (Current and ın Development) 35

2 3 Summary of the process modellıng methodologıes data they represent 67

3 1 summary of some of the avaılable and underdevelopment standards for constructıon 99

5 1. Summary of soft vs hard research dıchotomıes 134

5 2. Summarıes of the data collectıon and analysıs methods and the purposes 159

6.1 Shared constructıon process specıficatıon requırements and correspondıng PSL representation 191

7 1 AutoCAD Desıgn output data representatıons 205

7.2. AutoCAD Design file formats 206

7.3 AutoCAD applıcatıon ımport /export and file save types/formats 206

7.4 Constructıon Computer Software (CCS) applıcatıon ınput/output data types 207

7 5 CCS ınput/output file formats 208

7 6. Input/output data types for Microsoft proȷect scheduling applıcations 209

7 7. Mıcrosoft project schedulıng file save and file open formats 210

8.1. Metal buıldıng openıng construction desıgn detaıl 216

8 2 Metal buıldıng openıng constructıon desıgn detaıl 217

9 1 (a) PSL parts and theorıes used in the ınformation translatıons process to capture the AutoCAD design related process informatıon 289

9.1 (b). PSL parts and theories used ın the ınformation translatıons process to capture the CCS cost estımatıng related process ınformatıon 290

9.1 (c). PSL parts and theorıes used ın the ınformation translatıons process to capture the Mıcrosoft proȷect scheduling related process ınformatıon 291

List of Abbreviations

AEC/FM	Architecture Engineering Construction and Facility Management
AutoCAD	Computer Aided Drafting tool
BOM	Bill of Materials
BoQ	Bill of Quantities
CCS	Construction Computer Software
CDF	Comma Delimited Format
CIT	Construction Information Technology
DFD	Data Flow Diagrams
DWG	AutoCAD Drawings
DXF	Data Exchange Format
EDAPS	Electromechanical Design and Planning System
IAI	International Alliance for Interpretability
ICT	Information Construction Technology
IFC	Industry Foundation Classes
ISO	International Standards Organisation
IT	Information Technology
KIF	Knowledge Interchange Format
Macro OP code	CCS Macro Operations Code
MS Project	Microsoft Project tool
MSID	Manufacturing Systems Integration Division
NIST	National Institute of Standards and Technology
OP code	CCS Operations Code
PIF	Process Interchange Format
PSL	Process Specification Language
SDF	Space Delimited Format
STEP	Standard for Exchange of Product Model Data
UML	Unified Modelling Language
WfMC	WorkFlow Management Coalition
XML	Extensible Markup Language

Chapter 1 Introduction

1.0 Introduction

The aim of this chapter is to provide the background for the research work presented in this thesis. The first section describes the general context of the research with a focus on opportunity for improvement and advancement in construction Information Technology (CIT) The second section defines the problems and highlights the goal of the research while the third section justifies the research study. In section four the methodology used in this study is described Section five defines the scope and the specific context of the project. Finally, the structure of the thesis outlined

1.1 Background

1.1.1 Construction Projects and Processes

The successful completion and management of construction projects is entirely dependent on the degree of collaboration and coordination that can be achieved among the several participants who usually come from diverse organisations. However, effective collaboration is often hindered by the diversity and complexity of the construction processes and information handled during projects and the difficulties in exchanging this information.

Traditional approaches to collaboration have proved ineffective and are sometimes responsible for quality problems and the increased cost and duration of construction projects. For this reason, the need for effective computer-based collaboration between participants in the construction industry has become a major issue for researchers and standardisation bodies.

The increasing cost, duration, and quality problems in construction projects resulting from inefficient collaboration and poor methods of communications have been around in the construction industry for sometimes. Efforts by industrial and academic researchers and standardization bodies are still underway to find a mechanism for overcoming this problem. In order to overcome this problem the industry has to rely on the development and use of advanced information and communication technologies supported by information standards for defining and exchanging information. However, standardization has been slow in construction compared to other counter parts such as the manufacturing industry, this is due to the complexity of construction projects and processes, the one off nature of construction projects, changing environments and the fragmented nature of the industry.

Construction is an information intensive field (Ma, Z. and Chen, J. 1999) yet it is unique in that the information is specific to each project. This information is generated and handled by the many participants, who usually come from several diverse organisations using different technology applications throughout the project lifecycle. The successful management of the overall construction processes is entirety dependent on this information that has to be consistent, up-to-date and standardized The bottom line is that IT applications for construction processes have to support a continuously changing environment with complex and extensive information needs. The computer industry has provided the necessary basic technologies that would facilitate the development of new tools suitable for construction processes.

1.1.2 IT supports for Construction Processes

Information in construction organisations covers the different aspects of projects such as products, resources, and processes and is shared by participants throughout the stages of the construction process lifecycle. These include designers, project managers, contractors, suppliers, clients etc. In this research study the focus is on the exchange of (process aspects of construction projects' information) process information handled in the pre-construction process stages, which involve design, cost estimating and scheduling or planning between the various participants· designer, cost estimator and scheduler. The term "process information" refers to the information describing a process needed to realise a project product including, amongst others, a high level description of the activities, data and resource requirements, ordering relations and temporal constraints, cost etc. This information is generated and handled by the different participants during the pre-construction stages of projects using different methods of information representations and technology applications. However, during these processes the completion of each of the participants' task is dependent on data obtained from other participants. During or after the completion of the pre-construction processes, the information generated is required for the management of the construction projects. Hence there is a need for communication and flow of information between pre-construction and construction process stages. The effect of all this is, an increased need for integration of the processes and process information and effective collaboration between participants.

With the wide deployment of computer applications in Architecture, Engineering and Construction (AEC) organizations today, the majority of construction information is produced electronically using IT applications such as CAD systems for drawings, word processors for textual specifications and spreadsheets for tabular data and diagrams for graphical representation of project information. One of the primary areas in which computer applications are highly used to support construction processes are the pre-construction process stages of projects. For example, the design, cost estimating, and scheduling and planning information can be generated using any of the software packages available for construction such as AutoCAD, Construction Computer Software (CCS), and Microsoft project. However, the exchange of this information electronically can be very difficult. Solutions are needed to standardize information representations within software applications Currently, information is still exchanged via paper documents, individual computer application files and disks (IT in Civil Engineering, 2003).

Efforts have been made to develop better tools for creating and storing information on one hand and standardising information representation structures on the other. The focus of these efforts has been on specific domains. For example, the standardisation of CAD layers (ISO TC 10, 1993) through the attachment of semantic meanings, which is aimed at supporting a complete exchange of meaningful and understandable information among CAD users only. The classification and categorisation of construction projects information (ISO 1994a) without the definition of semantics is another example. However information in construction needs to be exchanged and shared between different applications such as CAD, cost estimating, and scheduling etc. hence there is a need for wider standardisation of information representations.

1.1.2.1 Current Construction Information Technology (CIT) practices

As described earlier, the whole "body of construction information" encompasses different aspects of projects such as products, processes, resources etc., and each of these aspects are viewed differently across domains throughout construction projects' lifecycle. For example design, cost estimating, and scheduling or planning views of the processes, products, and resources in the architectural and project management domains. Each of these aspects and views has unique information representation formats and structures, and terminologies depending on the domain of origin and use of technology applications. For instance, for the design, cost estimating, and construction scheduling information representation different sets of conventions are used. For the design of project products, the geometrical representation is used as the basis. The information embedded in the geometrical representation can be conveyed by means of drawings, structured according to industry and technology applications specific conventions In addition to this, textual information can be added to the drawings or collected as separate specifications of requirements, material and workmanship. For cost estimating information of the designed products, a tabular representation is used. Also for the designed products' realisation or cost estimating activities scheduling information tabular and graphical methods of representations are employed.

As described earlier, with the wide use of computer applications in construction to day, majority of information is electronic. However, electronic information although well represented, easy to edit and structured, there are difficulties to exchange this information without a "neutral information representation structure". The complexity of construction projects and processes and the interdependence of the process stages resulted in a growing need for information exchange. As a result, the development of mechanisms for context independent representations of information and structuring of the information representations and standardisation of the structures have become the focus of a lot of research for academic, industrial and standardisation bodies.

Several groups have been working on various aspects of standards for context independent information representations of to facilitate the integration of heterogeneous software applications for the construction industry (Thomas Froese, 1994). As a solution to some of the integration problems in the construction industry there have been some developments These are; the introduction of standardised classification systems for construction project information (ISO TC59, 1993); a major improvement in the structuring of CAD data has been the introduction of layering conventions, the International Organisations for Standardisation has defined international CAD layering standards for construction (ISO 1998b), and the introduction of Electronic Data Interchange (EDI) that supported the exchange of data such as CAD files and Bill of Quantities (BoQ) using agreed upon standards of data formats. However, this work resulted in the need for an advanced approach and methods for information standardisation.

In contrast with the above approaches, the introduction of advanced methods and approaches such as the conceptual modelling methodologies and product data technologies have exhibited great potential and promising solutions for an efficient way of representation and exchange of information hence an increased software interoperability (Vernadat F.B., 1996).

Product modelling techniques have been identified by many researchers as well as standardisation organisations as the key technology for solving software interoperability problems (Jagbeck A. 1998) Two main international efforts, the STEP and IAI have been engaged in the development of standards for conceptual schemas (structure of product models) for product information exchange. The ISO STEP standard (Standard for the Exchange of Product Model Data, ISO 10303) (ISO/TC184/SC4, 1993a) and the International Alliance for Interpretability (IAI) IFCs (http://www.iai-international.org/iai_international/) are aimed at product information exchange in the AEC. These efforts have demonstrated good potential of the product model approach for product data and product data related information exchange between heterogeneous software applications.

Internet technologies are influencing perceptions and attitudes in information communication technology (ICT). The expectation of access to shared information and communication is drastically different from what it was a few years ago (Froese, T.M., and Yu, K.Q., 1999). Currently, communication via e-mail and document sharing via the WWW are well established practices. Increasingly, the Internet will be used for distributed systems and generic data exchange. These technologies, in conjunction with information representation standardization developments, will lead to richer, more industry-specific data exchange methods. This is important for the construction industry where information is complex and communication is highly important and will lead to information integration and AEC/FM software interoperability.

Despite all these efforts and developments in information standardisation for product data integration and exchange the process or activity centred information of construction projects have not received enough attention. Therefore the aim of this research study is to identify and develop a mechanism for process centred information integration and exchange across different stages of construction and between construction software applications focusing on the pre-construction process stages of projects.

1.1.2.2 Information in the Pre-Construction Process Stages

The aim of the efforts described above such as the STEP and IAI for AEC is towards defining standards for the representation and exchange of product data and product data related information between all kinds of AEC/FM applications. One example is the IAI project management effort for production planning (Froses, 1998) with a focus on product related process information. In these efforts product data technology is employed for the representation and structuring of all the necessary information that are required to describe a product from product and process point of view hence provide the basis for the standardisation process. Information concerning products is given meaning by formatting it as a product model. The models include representations of product items and relations between them. Additionally, processes and their relation to product data have been added to the models (IAI 1997), an example of this representation from IAI is shown in **figure 1.1.** The figure shows an example of a conceptual schema for construction, a simplified part of the IFC 1.5 schema defining the relations between products and processes. One of the benefits of product modelling is in the transfer of information downstream, to the construction process (Tarandi Väino, 1998). The semantics of the product can be connected to the semantic structures in other process-related procedures such as cost estimating, production planning and scheduling. This indicates that the product models of these efforts don't support information representation related to the construction of project products such that the construction process information is not included. In order to facilitate the exchange of product and process information between all types of applications including process related software applications, the product models in these efforts, which are based on product data technology may need to consider the representation and structuring of product as well as process information i.e the product model need to incorporate process related information in addition to the product data and product data related information.

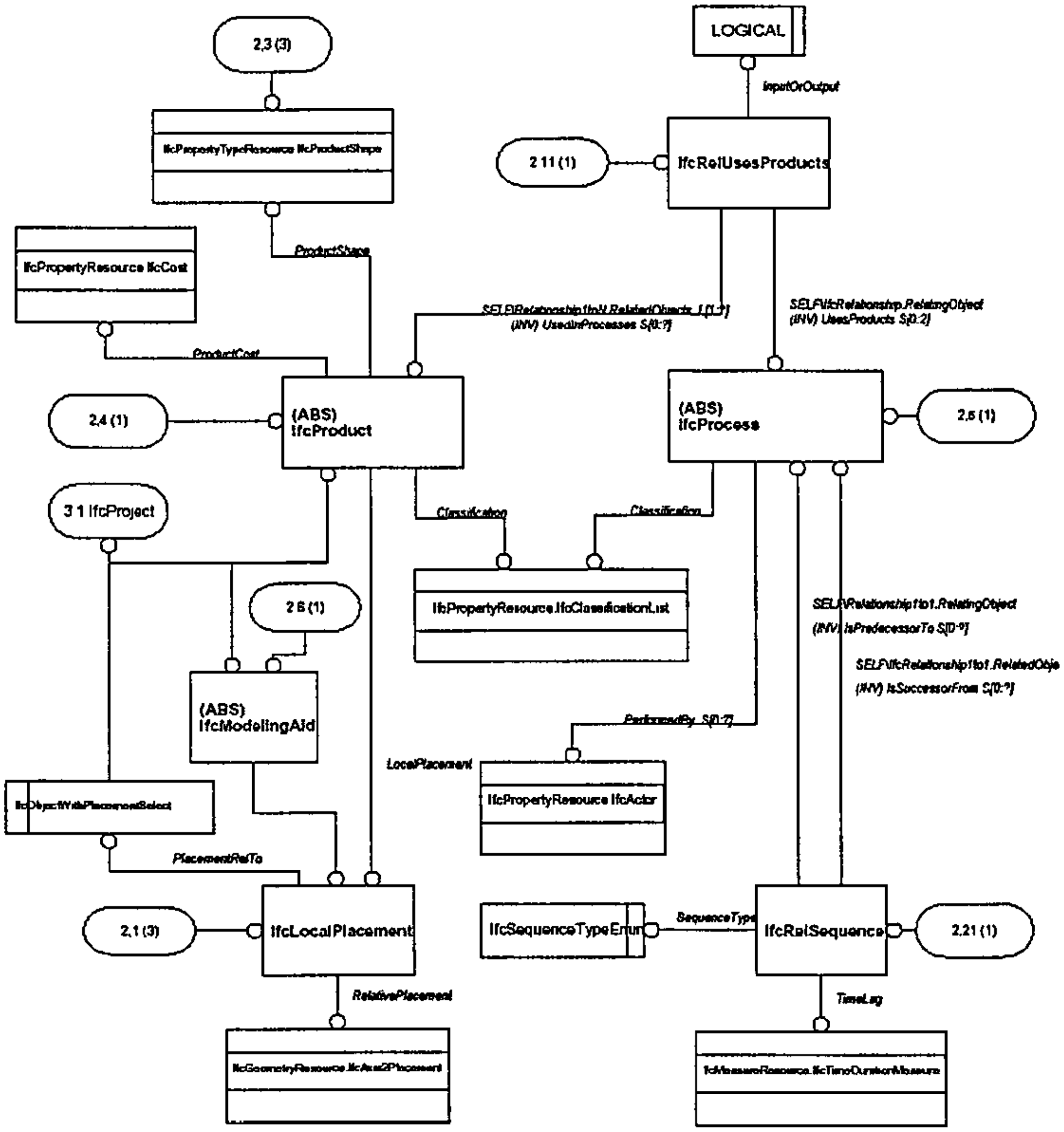

2,3 (3)
IfcPropertyTypeResource IfcProductShape
IfcPropertyResource IfcCost
ProductShape
ProductCost
LOGICAL
InputOrOutput
2 11 (1)
IfcRelUsesProducts
SELF\IfcRelationship1toN.RelatedObjects L[1:?]
(INV) UsedInProcesses S[0:?]
SELF\IfcRelationship.RelatingObject
(INV) UsesProducts S[0:2]
2,4 (1)
(ABS)
IfcProduct
(ABS)
IfcProcess
2,5 (1)
3 1 IfcProject
Classification
Classification
IfcPropertyResource.IfcClassificationList
2 6 (1)
(ABS)
IfcModelingAid
SELF\Relationship1to1.RelatingObject
(INV) IsPredecessorTo S[0:?]
SELF\IfcRelationship1to1.RelatedObje
(INV) IsSuccessorFrom S[0:?]
PerformedBy S[0:?]
LocalPlacement
IfcPropertyResource IfcActor
IfcObjectWithPlacementSelect
PlacementRelTo
2,1 (3)
IfcLocalPlacement
IfcSequenceTypeEnum
SequenceType
IfcRelSequence
2,21 (1)
RelativePlacement
TimeLag
IfcGeometryResource.IfcAxis2Placement
IfcMeasureResource.IfcTimeDurationMeasure

Figure 1.1 IFC 1 5 schema defining the relationships between products and processes (IFC 1.5, IAI 1997)

In these models the integration of information is based on the connection between the processes and product data as shown in **figure 1.1** this is based on data requirements for these processes, and information related to the requirements of the processes in producing the products. Generally, the product models are rich in describing products but lack some constructs to describe the processes, the IFC 1.5 schema of **figure 1.1** illustrates this However the IFCs included some project management related objects in the IFC Release 2.0

In addition to physical information about buildings, the Industry Foundation Classes (IFCs) tend to represent project management information such as estimating and scheduling data. In addition to the core concepts relating to the project management portions, the IFCs have included some project management-related objects in the IFC Release 2.0. In February 1999 the Project Management (PM) Domain Group of the IAI North American Chapter (IAI NA PM) held a workshop to test how well the IFCs represent the cost estimating and scheduling information (T. Froese et. al., 1999). From this workshop exercise it was learned that the need for changes of class names between versions (from IFC Release 1.5.1 to IFC Release 2 0 Beta) became inevitable and were confusing and required redundant work during the implementation of the PM test case.

Therefore, it is difficult at this stages to argue that the product model data representation could facilitate the exchange of the "collective body of a construction project information" including the "process information". This is because the product modelling methodology does not provide the facility to represent the entire set of requirements necessary for specifying construction processes hence the product related process information represented in the product models in these efforts is not complete enough to represent the construction processes completely. For process centred project information representation and exchange suitable approaches and techniques such as process modelling and representation methodologies that would include the representation of process specification characteristics are required. Process specification should include the notion of sequence, constraints, data requirement, time duration, resources and resources related information etc. To date, no literature has shown a single product model that incorporates all the necessary information that can provide a complete specification of processes. Process modelling or formal representation of process related information using formalised modelling methodologies could complement product data technology approaches. To facilitate a formal and structured representation of process information and to provide a basis for a common definition of the information, a suitable modelling methodology that can represent all the necessary requirements for specifying processes is required. Among the available process modelling methodologies, IDEF3 Process Flow and Object State Description Capture Method (Knutilla, A., et. al., 1998) has been identified as the best modelling methodology for process information representation. This approach provides the best coverage for the representation of information requirements, which are necessary for specifying process

1.1.2.3 The Need for an Agreed Interpretation of Process Models

Process models are formal representations of collective set of interrelated activities, and participating objects structured using some formalism (or language) defined by constructs of a modelling methodology and by the syntax and semantics of the methodology. One of the objectives (Curtis, Bill, M. I. Kellner and J Over, 1992) of process modelling is to facilitate interaction among groups working in different phases of a process. This can be achieved only if a model is capable of providing an understanding or a neutral representation of the information exchanged between the groups. However, almost all existing, process modelling methodologies focus on the syntax of process specification rather than the meaning of terms, the semantics (Knutilla, A., et al. 1998), while this is sufficient for exchanging information between applications of the same type, different types of applications associate different meaning with similar or identical terms. Communication between heterogeneous applications is more concerned with the semantics of the information exchanged rather than the terminology and data formats. An important requirement for semantically integrated applications environments is the ability to capture information from multiple domains/applications and define this information in a form that can facilitate a correct information sharing. There is therefore, a need for an agreed interpretation of process models representations/terminologies i.e. a formal and explicit specification of the semantic concepts of the process model. This specification would facilitate the representation and exchange of accurate process information between heterogeneous process-related applications. One approach to generating this specification is through the use of ontology. Semantic definitions of models can be expressed in the form of ontology, i.e. using a neutral knowledge representation format (Neches et al , 1991).

Ontology (Uschold M., & Gruninger M., 1996) is an explicit specification of conceptualisation, where conceptualisation is an abstract view of the world represented as a set of objects. Ontology is a formal description of terms, entities, objects, and classes and their characteristics, relationships, constraints and behaviours (Uschold, M.

and Gruninger M., 1996), and provides formal definitions and axioms that constrain the interpretation of terms (Gomez-Perez 1998). It can also be seen as an explication of the context for which a term is normally used Ontology provides a sharable representation of knowledge that is a shared terminology that captures key distinctions among concepts in different applications and defines the meaning of each term in a precise and unambiguous manner to facilitate translation of concepts among these applications. The advantage of the use of ontology is the ability to overcome the problem of implicit and hidden knowledge by making the conceptualisation of a domain explicit. This is the focus of this research study, which, studies ontology and ontology based developments and their applications in bridging the semantic gap that prevailing between various information representations and construction software applications.

1.2 Problem definitions and research goal

The previous section, highlighted the obstacles to applications software interoperability and shown that the product data technology has been identified by many researchers as well as standardisation bodies as a key method for resolving this problem Numerous groups working on information standards that aim to facilitate the exchange of product data and product data related information of construction projects have employed this technology. Among these the main two efforts: the ISO STEP (ISO/TC184/SC4, 1993a), and IAI (http //www.iai-international.org/iai_international/) have produced provisional standards and prototype tools demonstrating the potential of the product model approach (Jagbeck A. 1994). However, the integration or standardisation of product related information of construction project alone couldn't bring a complete solution to the software interoperability problems of the sector for two reasons: construction project information includes different aspects of projects including "construction process", and the other is, there are numerous software applications that deal with the manipulation and management of process information and their interoperability is vital in the industry. In the previous section it was shown that processes and their relations to product data have been added to the IAI's product models, however these relations are bound to the data requirements of the processes alone, while there are large amount of process related data that need to be manipulated and shared amongst process related construction software applications. Hence there is a need for a methodology with a potential to resolve the process related information exchange problems, which hinders the interoperability of process related construction software applications that deal with process centred information in projects.

To date, there are a wide range of software applications, which deal with the management and representation of construction process information. Examples include AutoCAD for design, Construction Computer Software (CCS) for cost estimating, and Microsoft project for planning and scheduling, each of which is developed and supported by different programming and control languages. Each of these applications focuses on particular aspects of a process and uses a unique representation and terminology to manipulate and represent process information. For example the description of a typical process should include the notion of the data requirement, product to be constructed and the materials of construction from design, resource requirements and cost from cost estimating and time-duration, dependency etc. from scheduling and planning applications. However, the meaning of the terminology representing process information in any of these applications is entirely dependent on the context in which it is viewed and interpreted For this reason, electronic information exchange between the applications is hindered. In order to reuse information generated at one process stage using a specific application as input in other applications rudimentary methods such as communication via paper documents, individual computer files or verbal communication are still used. This is acute in the construction sector because of the growing complexity of construction information and the increasing need to distribute and share this information amongst construction project stages, participants and supporting software applications. Nevertheless, an effective, electronic transfer of information between the applications could reduce the efforts of obtaining information and the risk of losing or distorting it at the destination. One of the primary obstacles to such environment is the lack of a "neutral language" or formal specification of the underlying concepts of processes and a common representation of process information in the domain A formal specification of the construction process description would be a rigorous foundation to the solution of the problem

The National Institute of Standards and Technology (NIST) have identified the development of Process Specification Language (PSL) (Schlenoff, C., Knutilla, A., Ray, S., 1996) as a key mechanism for process information integration and process related software applications' interoperability in the manufacturing environment. Motivated by a growing need to share process information in the manufacturing environment, PSL is a project undertaken by the National Institute of standards and technology (NIST), aimed at developing a generic Process Specification Language, as a language for the description of processes and as an interchange language for process information exchange between heterogonous process related manufacturing applications, building on existing modelling methods. In this research study, it is envisaged that in an analogous manner to the role of PSL in the manufacturing environment, construction organisations could benefit from a similar standard for the exchange of process information between process-related construction software applications

1.2.1 Aim and Objectives of the Research Study

Based on the problems and the potential that some emerging technologies have in overcoming such problems as discussed above the overall aim of the research is defined as follows:

To explore the applicability of Process Specification Language (PSL), developed by the Manufacturing Systems Integration Davison at National Institute of Standards and Technology (NIST), in the construction industry, and to trial its use through the implementation of the language in a construction case study scenario.

The Research questions are:

- *How applicable is PSL in construction environment?*
- *Provided proved applicable, what is the impact of PSL in construction data interoperability?*
- *What is the impact of construction process concepts on the use of PSL in construction?*

The aim of this project is achieved through the following objectives; this will be further discussed in the following section within this chapter.

The objectives included are:

1. Study and analyse PSL and its process concepts
2. Identify an appropriate construction scenario relevant to the PSL objectives, and gather process information representations, that are necessary for modelling construction processes within the scenario, through technical analysis of software

applications supporting the scenario and to determine if there exists a common set of information representations between the software applications

3. Develop process model for construction of a case study project, using IDEF3 modelling technique, and identify and collect process data within the scenario i.e. case study project construction process information related to each of the software applications supporting the scenario for each activity in the process model
4. Exchange the construction process information of the case study project between the scenario software applications using PSL as an interchange language between the applications information representations.

1.3 The Research

The research problem was investigated at technological level in the following areas: process technology, information technology, and communication. The research focused on issues related to: construction information technology (CIT), context of IT in the construction organisations, IT tools and support for construction, current practice of IT in construction, construction organisations and processes, method of communications and difficulties.

The aim and objectives of the research was achieved through the following phases:

Literature review: the aim of this phase was to provide the basis for the research work through a review of the construction projects, processes, participants and information technology (IT) in the industry; existing methods of collaboration between construction projects participants with a focus on software interoperability; academic and industrial research and developments underway for software interoperability and the technologies and methodologies used. Previous and ongoing research and developments for software interoperability and information integration including the methodologies and technologies used, and the proposed candidate solutions were reviewed. Standards and standardisation efforts and developments for software interoperability for different aspects of industrial projects, Product and process modelling methodologies, and developments related to process and product models were also reviewed.

Identification of Construction Scenarios: this phase involved the identification and definition of a construction scenario relevant to PSL objectives, and the selection of software applications for the scenario. A scenario within the pre-construction process stages, which involve design, cost estimating, and scheduling information representation and exchange was selected as a case study scenario for the first trial implementation of PSL in construction. Software applications, which deal with process-related information, in the pre-construction process stages of projects, have been identified and a selection of a number of these has been made to determine candidate software applications for the scenario. These are: AutoCAD, Construction Computer Software (CCS) and Microsoft project

Analysis of the Construction Scenario Software Applications: in this phase the software applications selected for the construction scenario were analysed to identify and gather the data types necessary for the applications to represent processes information. The process related data types of the software applications were analysed to determine a common set of data types, which represent a shared semantic concepts between the applications that are necessary for specifying construction processes.

Selection of Case Study Project for the Construction Scenario: in this phase, part of a building project was selected as a case study project, and design, pretender cost estimating, and scheduling related process information were identified, based on information gathered from the construction specification document of the case study project in correlation with the scenario software applications data types.

Development of Process Models: in this phase two types/sets of construction process models were developed, within the construction scenario using an IDEF3 Process modelling methodology, these are:

(1) *Architectural design, pre-tender cost estimating and scheduling information representation processes models:* The scenario software applications' users tasks or activities were represented/modelled for building projects in general

(2) *Construction process model for the case study project*: In this phase, a process model for the construction of the case study project was developed and information related to the scenario software applications were collected for each activity of the process model. This model represents process information of the case study project related to the scenario software applications, structured in terms of the modelling methodology used.

Process Information Representation using Scenario Software Applications: in this phase the construction process information of the case study project is represented using the scenario software applications information representations methods.

Definition of the Construction Scenario Process Concepts: in this phase the native ontologies of each of the scenario software applications were developed using the data types that were gathered during the analysis of the software applications and the corresponding process data of the case study project.

Implementation of PSL in the Construction Scenario: in this phase, construction process information of the case study project was exchanged between the scenario software applications, suing PSL as interchange language. This involved development of hand written translators (translation definitions) between the scenario software applications and PSL, and a manual process information translation between the scenario software applications using the hand written translators (translations definition) developed between the applications and PSL.

Evaluation of the impact of PSL in construction: the effective use and impact of the PSL language in construction is evaluated through an analysis of the implementation exercises.

1.4 Justification for the Research study

This research is important on several theoretical and practical grounds.

- The quality, duration and cost of projects are important considerations within construction organisations. Construction projects encompass intensive and cumbersome processes involving numerous participants, experts and professionals throughout a project's life cycle. It is always the combined efforts of these participants that achieve the planned goal, the completed project, and their collaboration is important in achieving a quality product within minimum time duration and cost. Hence, there is a need to improve collaboration between construction project participants and promote innovative techniques. This led to an increased need for the development of modelling concepts and standardisation of information representation structures particularly in terms of how information technologies are used and could be used to support innovative collaboration and coordination.
- Information intended for use in construction projects is mostly produced on computers using IT tools and computer applications, such as CAD systems for drawings, word processors for textual specification, spreadsheets for tabular data and charts for graphical information. Hence it is these applications and tools that provide the basis for the representation and managements of the information generated by

construction professionals, and their interoperability is cornerstone to an improved collaboration between the participants. Though the construction is a late adopter of advanced Information Technology (IT), the role of IT in the adoption of innovative (computer based) collaboration is under great scrutiny. The product data technology has been as a potential and key technology in the information standardisation developments and researches for software interoperability. Based on this technology, there have been several initiatives underway to improve software interoperability and hence improve collaboration. These initiatives have been working to develop standards for information exchange based on product models to support the exchange of product data and product data related information within heterogeneous software applications. However the process centred view of construction projects information has not been receiving enough attention. The applicability of the process description techniques such as the "Process Specification Language" (PSL) of NIST for exchanging project information in construction has been studied (Gruninger, et al. 2003) however the use of PSL in construction requires further investigation particularly in the applications areas of construction process information management. The application of PSL in the construction environment could complement the product data technology approaches being explored and support the efforts made for complete interoperability of applications software in construction.

- The adoption of IT by construction organizations has become commonplace in today's specialised industry. From everyday tasks such as word processing to advanced tasks such as structural analysis, IT has supplanted other methods to become the standard operating procedure for the industry (IT in Civil Engineering 2003). However, this rush for automation often occurs with little regard to measurement of success. One of these is the ability of IT tools and computer applications to interoperate and exchange meaningful information. As described above, information exchange can be difficult because of lack of standards or common languages to facilitate the exchange. There are some commonly used de-facto standards such as DXF, which can be used for information exchange between CAD systems and other spreadsheet based applications. However, although sufficient for CAD systems this doesn't support the semantics of the information exchanged between CAD and other applications. As a result There is a need for those standards to be replaced by higher level ones with far greater semantic content or a further development of IT tools to support an accurate and complete exchange of information between heterogeneous applications.

- The investigation of the applicability of PSL in construction and the implementation of the language in a selected construction scenario are conducted within an academic context. However the software applications supporting the selected construction scenario are used to reflect the various participants involved in construction projects within the scenario. The information generated, handled and exchanged using the computer applications also represents a real construction project scenario.
- The findings of this study will be useful for researchers and developers interested in the area of process information integration and exchange and software interoperability. Commercially successful standard products of general use throughout the industry are unlikely to be produced if research projects continue to take individual approaches to the definition and implementation of basic objects relating to buildings (IT in civil Engineering 2003). The research study findings will provide the basis for comparison between the initiatives for applications software interoperability based on "product data models" such as STEP and IFCs for product data exchange and "processes description languages" such as PSL for process information exchange. Hence, this will be useful in the determination of whether the initiatives based on product data technology such as the STEP and IFCs can be manipulated to incorporate process specific concepts to the product model and clear the road to single successful standard production for construction organisations.

1.5 Research Issue and Context of the Project

In the department of Civil and Building Engineering of Loughborough University the potential of advanced Information Technology (IT) for construction projects information management and innovative collaboration between construction project professionals have become one of the major areas of research.

As construction projects become increasingly complex with many stages and participants involved each with different, culture, and computer programs, collaboration between parties become harder to realise. One of the main problems is that almost all computer applications are used in a stand alone manner each with its own internal data representations; data formats and language, which makes communication between them, difficult without application specific translators. Different construction functions and participants including their computer applications may use different terms to mean the same concepts or same terms with different meanings. This has resulted in the misinterpretation of the concepts of the message exchanged.

This research study aimed at developing an interchange language based on Process Specification Language technology for process information specification and process related software interoperability within construction projects. Where there is no live industrial construction case study in a research such as this, the computer applications supporting various construction process stages are used to represent the numerous professional disciplines involved in construction processes and use a real construction project context. This includes the information generated, handled and exchanged between the participants throughout the stages of construction project lifecycle.

Based on a comparison between the current collaboration practices and the possibilities offered by advanced Information Technologies (IT) for innovative collaboration, it can be concluded that the use and development of advanced IT holds greater promise for interoperable computer applications and integrated construction IT environment.

1.6 Structure of the Thesis

This thesis is organised into five parts. The **first part** introduces the research problem and highlights the goal of the research and justifies the research study in **Chapter one.** The **second part** presents research issues in areas relevant to the research problem; reviews pervious and current developments for industrial information integration, software interoperability, and the information modelling methodologies employed. The **third part** reviews some of the available research methodologies for IT related problems and describes the overall research and methods used in this study. The **fourth part** presents analysis of the research problem and results of applying the research methodologies in this research while the fifth **part** presents the discussions, conclusions and recommendations about the overall research work based on the results from part four.

Part Two: literature Review in Areas Relevant to the Research Problem

The **second chapter** reviews the two main information modelling approaches (product and process modelling) including the background and objectives, and modelling concepts of these approaches; and a number of the available process representations, methodologies, and languages. It studies the strengths and weaknesses of these methodologies for modeling process characteristics that are necessary for specifying construction processes and for integrating different applications. **Chapter three** reviews information standardization developments and efforts for Architecture Engineering and Construction (AEC) and identifies, the product modeling has been the major approach in these developments and while most of the existing standard and ongoing developments and efforts focused on information standardization for product aspects of construction projects information exchange the process centered of construction projects have not received enough attention to date. The **fourth Chapter** reviews the ontology and process specification technique based information standardization approach for process related manufacturing software applications interoperability and studies the possibility for adapting this approach to the problem area of process information exchange between software applications in the construction environment.

Part Three: Research Methodologies in Literature and Methods Used

Chapter five reviews some of the available methods of data collection and mode of analysis for IT related researches It identifies the theoretical framework of the research and methods of data collection, and model of analysis used.

Part Four: Analysis of the research Problem

To provide basis for the trial implementation of PSL in a construction scenario **chapter 6** presents an investigation of the applicability of PSL in construction, and identifies the PSL parts that contain applicable semantic concepts for defining construction process information representations within the pre-construction process stages (process related AutoCAD design, CCS cost estimating, and Microsoft project scheduling applications data types) and the need for new extension developments to incorporate the construction process concepts that are new to the PSL ontology. For the trail implementation of PSL in construction **chapter 7** presents a detailed description of a construction scenario within the pre-construction process stages including descriptions of the process information representation processes within the pre-construction process stages, the scenario software applications selected to support these processes in these stages, a brief description of onsite construction process and an example of a situation within the scenario.

To limit the scope of the construction process information exchange between the scenario software applications, the **first part of chapter 8** identifies a case study project (part of a building construction project) for the construction scenario defined in chapter 7, and presents a detailed description of the case study project using IDEF3 Process Description Capture Method and the scenario software applications' process information representations. The **second part of chapter 8** presents the result of the case study project construction process information exchange between the scenario software applications using PSL as an interchange language.

Part Five: Discussion, Conclusions and Recommendations

Chapter nine analyses and discusses the combined results of the research, while **chapter ten** presents conclusions and recommendations based on the results of the overall research.

1.7 Delimitation of the Scope

The study is focused on the use of the PSL standard language for process information exchange between software applications in construction that could compliment the product model data standards for product data exchange. The capability of the product model data standards to support the exchange of the whole set of construction projects information including the construction process concepts that are necessary for specifying construction processes was not investigated.

Chapter 2 Construction Information Modelling and Integration

2.1 What is Construction Information?

The foundation to answering this question lies in the identification of the “domain of information” and “subject of information” involved in construction. For example in the domain of architecture the subject of any piece of information is an architectural artefact or some kind of architectural concepts. Similarly in the domain of project management: the subjects of information are products or activities and resources for cost estimating, and activities and resources for scheduling and planning In such construction domains the resulting information is not know in detail before the definition of information has been completed. Additionally, these information defining the subject of information in any of the construction domains are often ill structured, inconsistent, incomplete and may be subject to changes during the production of the information or after and the exact meaning of the terms may not be unambiguously defined. Traditionally information in an organisation exist in the form of organisational charts established by management, documented operational procedure, regulation texts, and to a large extent in the vast amount of enterprise data (either in databases, knowledge bases, or simply data files) and code of application programs Moreover, a large amount of information remains in the mind of people and is not formalised nor even documented at all. Although, a wide range of computer applications have been introduced in construction stages to improve the traditional form of information management, that resulted in a well-structured and digital information representation (physical file formats), information exchange using these application is still inconsistent. As a result formal representation of information are needed in order to support consistent information production and facilitate information exchange using computer applications.

2.2 Information Categorisation and Organisation

2.2.1 *Subject of Information*

The primary means of information categorisation are subjects of information The subject of any construction information depends on the domain of the information and purpose and use of the information. For example in the architectural domain the artefact and in the project management domain the activities or processes and resources are subjects of information for the purpose of understating or modelling the architectural design data and project management information respectively Construction information is large, and complex and in order to achieve some comprehension of the information a distinction between subjects of information in each domain of information need to be understood and drawn clearly **Table 2.1** shows construction information at different levels of abstraction. The following are some of the distinctions (van Leeuwen, J P and H. Wagter 1998) that can be found in many disciplines of applied information and communication technology (ICT):

First different levels of abstraction can be distinguished for example in the architectural domain. Information is related to complete buildings, constituent parts of a building, components, spaces, materials and so on.

Second: distinction can be made between physical part of building and the non-physical concepts that play a role in design.

Third: distinction is between information that is related directly to the buildings and information that is related to the context of the building for example, information concerning the building site or its environment

Fourth: important distinction is between project bound information and information that is not related to a particular design case or building, but it is common or represents general knowledge Examples are product information, material properties, physical principles, normative regulations and building codes

Information	**Example**
Project information	Building construction project information
Domain information	Architectural design/cost/scheduling data
Component information	Product or process data
User information	Product or process attribute

Table 2.1 Constsruction information at different levels of abstraction

2.2.2 *View, Aspects and Perspectives of Information*

Due to the diversity of construction information a complete representation of the whole would be too large and complex to be useful, prosecutable and even understandable either by humans or computer. For this reason it is useful for the information to be organised into groups Views, aspects or perspectives (Curtis, B., Kellner, M. and Over, J., 1992,) often coexist as mechanisms for structuring information.

The second common organisation of information is by distinction of view of information that is the views on the subject of information Views represent the information relevant to a domain. For example a subject in the architectural domain may have many views depending on the domain in which it is viewed and may be given different characteristics in other construction domains such as production planning, cost estimating, structural stability and so on.

The third categorisation of information is by aspects. Aspects of information in a domain are not confined to a particular discipline but are used to select similar types of information that may be relevant to several participants. Often aspects are defined as properties of the subject of information or a concept in a domain for example the cost property of a process or product, other aspects may not be regarded as clearly as properties for example planning aspects.

2.3 Objectives of Modelling Construction Information

From the application of information technology (IT) point of view many of the objectives for modelling information are common to all areas of applications Generally, the main objectives are to acquire less ambiguity, in the way information are represented and stored, and to improve the speed and quality of retrieving, manipulating and communicating information. One of the main objectives in modelling construction information is to improve communication among participants in AEC/FM. For improved communication it is important to understand the domain in which information is communicated It is also important that information is represented unambiguously, that is the information in the receiver domain matches the information concepts in the sender domain and vice versa. In order to achieve a better understanding of information between domains, information need to be organised in structured representation models. In summarising the problems of communication in transitional construction De Vries (1996) stated that communicating information on paper is a time consuming and error prone process, when parties copying information incorrectly, interpreting information differently or overlooking available information Although digital forms of information documentation tend to eliminate these errors the need for interpretation is still high This is because information in computer applications is implicit and not explicitly defined

2.4 Issues in Modelling Construction Information

Information modelling can contribute significantly to the processes of design and construction The IT industries provide the adequate tools for this purpose However there are many issues that are specific to construction industry compared to for instance other information technology applications and these issues require special attention From the construction information modelling point of view one of the main issues that need special attention is "integration of different views, aspects and perspectives of information" which is the focus of this research study.

Construction information is complex due to the many participants in the collaborative construction process. Communication and information sharing is one of the problems in the sector and brings along the difficulties in keeping information consistent across domains. Integration of construction information is important in managing these problems The challenge of integration is that views on information in domains to be integrated differ greatly. For example the view of an architect on a column on space as a spacing-disturbing element is different from that of a structural engineer in his domain viewing this column as a structural element transferring weight of itself and other structural elements above to another structural element below in a building. Additionally, different or the same terms may be used to represent information, in which the semantics of these differing or same terms are not explicitly defined. For example a "task" in planning domain is "operation" in cost estimating domain. Obviously communication between these domains would be confusing if a domain is not able to understand the exact meaning of the information from the other. Integration of information in construction requires common and formal representation and definition of the different aspects, views and perspectives and subjects of information.

2.5 Approaches to Information Modelling

Construction is mainly made up of products, process, resources and organisation entities, and construction information comprises a collection of the characteristics of these entities and relationships that may exist between them. These characteristics and relationships may represent different aspects and views for any subject of information However without an appropriate approach of information representation these views, aspects or perspectives may not be clear and obvious. Today there are numerous modelling methodologies that are designed and developed to represent different aspects views or perspectives of a subject area or information of a subject area Hence, as stressed above to make these distinctions clear it is important to follow/select an approach (information modelling approach) that would facilitate the representation of the required views, aspects or perspectives of the information based on the purpose and use of the information There may be several approaches to information modelling, however the product and process modelling are widely used approaches today and these are briefly described in the following sections.

2.5.1 Product Modelling Approach

Product modelling is an approach used to formally represent information related to a product in a manner that is aimed to cover the aspects of multiple disciplines involved with the product and in various life-cycle stages of the product. This approach has been identified by many researchers as well as standardisation bodies as a key approach for information representation and has been used in the main two initiatives, STEP (ISO/TC184/SC4, 1993a) and IAI 's IFCs (IAI 1997 and 1998) in developing information standards for product information exchange between applications.

2.5.1.1 Background and Objectives of Product Modelling

Backgrounds

The origin of developments in the area of product modelling is found in the need to create models that are semantically rich, and in the need to structure the integration of computer applications used in product development (van Leeuwen & van Zutphen 1994). Formal definitions of product data may be used to describe the semantics of products, resulting in models that relate semantics to the graphical representation of products to facilitate communication and exchange of information between computer applications There are numerous modelling methodologies and techniques that support this approach among these, the EXPRESS language (ISO (1994 b) which is one of the widely used methodologies is an object-flavoured information modeling language that was developed as part of the STEP product data exchange standard. EXPRESS was developed to enable a formal specification of the ISO standard 10303 Product Data Representation and Exchange

Objectives of Product Modelling

The objectives for creating product models (Eastman 1993) can generally be summarised as follows·

- Provide a formal description of product data covering the information requirements of all parties involved in the product design and production, during all stages of the design and production process.
- Offer the definition of formalised semantics for storage of information and enhanced access to information by a multitude of domains and computer applications.
- Improve communication between design and production participants by means of higher-level semantics of exchanged data and structured procedures for data exchange.
- Contribute to the development of standardisation of both data structures and exchange procedures

2.5.1.2 Concepts of Product Modelling

One of the major concepts in the development of product models is the notion of views Many different participants involve in design and production of a single product and each of these describes the product from different point of view, hence has a specific view on information related to the product In Architectural design, where the product is generally a building, this results in different views from participants such as the architect, structural engineer, HVAC consultants etc. each of these participants has a particular way of describing the information representing the product Moreover the set of information that is relevant to the different participants varies for each of them. These different views on building information lead to the definition of different information models for the particular requirements of each participant. In the traditional way of communication it is required that the individuals involved are able to read and understand the information from all those that take part in the communication. Computer-assisted communication requires that the digital forms of information can be read and understood by the software used on both sides of communication. This has been one of the problems in software development and standardization efforts in the area of CAD/CAM/CAE in the last two decades. (Jos. P. Van Leeuwen , 1999).

Product modelling efforts have incorporated the notion of views in the definition of information models and exchange methodology. This has been a subject of research in the AEC by, amongst others (Eastman, 1992, Rosenman, 1993; Van Nederveen and Tolman, 1992, and Amor and Hosking, 1993) Different kinds of models can be distinguished in this field of research Models that are defined for the purpose of a certain discipline, applications or view on a building are called view-models Information concerning a single building will be defined in different ways in different view-models for different purpose of applications. The fact that these view-models describe the same physical building is not modelled as part of the view-models One way of defining this relation between view-models is by means of a so called core-model. A core model could be regarded as a special kind of view-model, defining the information that is relevant to all or most of the involved participants in a way that is convenient or at least agreeable to all of them. This model should form the basis for the communication between the participants, meaning that it needs to contain definitions of the information that is to be exchanged between them. Figure 2.1 illustrates the concepts of product modelling (Eastman, 1992). STEP (Wix, J. 1996) models use this concept in representing a consensus view of information.

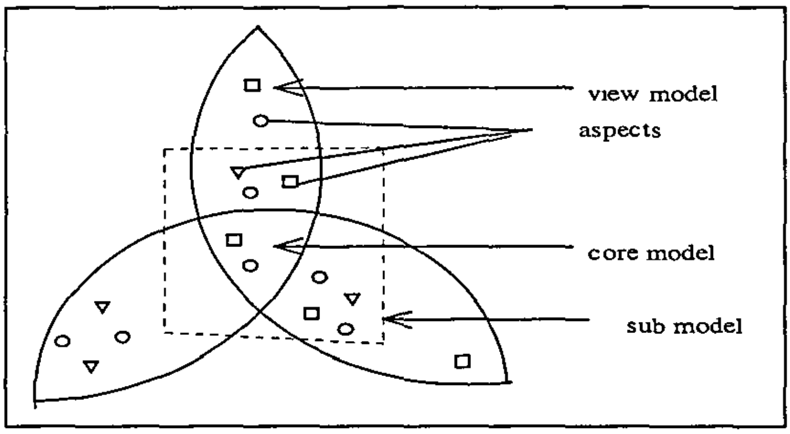


Figure 2.1 product modelling concepts (Eastman, 1992) view-models, aspects, a sub-model and a core model

2.5.2 Process Modelling Approach

Process modelling is an approach used to formally represent the information describing a process from multiple participant disciplines' point of view across different stages of products' development Process models describe activities to be carried out in the developments of products including the products to be created and the resources consumed and used including other information related to the product and other processes. This approach has not been used widely particularly in the construction sector compared to product modelling approach. However, there are developments in the manufacturing sector aimed to develop standards for process specification and exchange based on process modelling approaches These include A Language for Process Specification (ALPS) Project (Catron, B., Ray, S , 1991), the Toronto Virtual Enterprise (TOVE) Project (Fox, M , et al, 1996), the Enterprise Ontology Project (Uschold, M , et. al.,), the Core Plan Representation (CPR) Project (Pease, A., 1998), the Shared Planning and Activity Representation (SPAR) Project (Tate, A.,), the WorkFlow Management Coalition (WfMC) project (Belgium, 1994), and the Process Interchange Format (PIF) Project (Lee, J , et al,).

2.5.2.1 Backgrounds and Objectives of Process Modelling

Backgrounds

Historically, many of the results of process modelling available today have their origins in software life cycle and software development process In the 1970s it was clear that the production of computer based systems, and of software specifically, presented problems, which were not usually, present in more familiar "manufactured" products As computer systems became more complex and more pervasive, this contrast became increasingly more marked The first conference (Egham, Surrey, UK, February 1984,) on the "Software Process" was held in England in 1984 and has been followed by many others since. Various models of the software development process have been suggested and a number of modelling techniques developed, frequently associated with some form of computer support to provide assistance for software developers in following such development processes.

Objectives of Process Modelling

The process modeling may have four main objectives from (CIMOSA) projects point of view

- □ Acquire explicit knowledge about the processes
- □ Exploit this knowledge in process reengineering projects to optimize the operation
- □ Support the decision making activities
- □ Ease interoperability of the processes

2.5.2.2 Purpose of Process Modelling

The purpose of a process model is to dictate the elements of the process that are included in the model. Curtis (Curtis, B , Kellner, M.I , Over, J. 1992) identified the possible purposes of process modelling. understanding and communication; process improvement; enhanced process status visibility; automated process guidance; process enactment, and the reuse of successful processes These are briefly described as follows·

Facilitate human understanding and communication: process transparency helps people (including the customer) to communicate on the work to be done and to understand what part they play in the game. Process models may serve as a prerequisite for audits and as an excellent learning aid for employees. Requires that a group be able to share a common representational format

Support process improvement: process improvements are based on process model assessments, be it in terms of formal reasoning (e g. static or dynamic analyses) or visual assessment requires a basis for defining and analyzing processes

Support process management: a process model provides a sound basis for detailed planning and easier monitoring, measurement and co-ordination of an actual development project. Requires a defined process against which actual project behaviours can be compared

Automate process guidance: The documentation of process structures enables the capturing and later reuse of process know-how The objective is to provide "guidance, suggestions, and reference material to facilitate human performance of the intended process". Requires automated tools for manipulating process descriptions

Automate process execution support: This aims at the derivation of data structures for the development or adaptation of information systems supporting the enactment of processes Parts of the process can be automated, e g by distributing and supplying information and documents electronically with the help of a workflow management system. Requires a computational basis for controlling behavior within an automated environment.

2.5.2.3 Concepts of Process Modelling

There are many aspects that may form part of a process model. The particular aspects of a process that are modelled depend on the detail of how it will be used. Processes can be modeled at different levels of abstraction (for example, generic models versus tailored models) and they can also be modeled with different goals (descriptive models versus prescriptive models). Process models can be developed from different perspectives depending on the kind of information required Usually this of the type, what work is going to be done, who and how is it going to be done, when will it be done, who will take the decision. Hence, a process has functional, behavioural, organizational, informational, and decisional and resource based content. Based on the aspects to be modelled in their survey (Curtis et. al 1992) identified four most common perspectives to processes these are functional, behavioural, organizational and informational·

- **Functional:** *represents what process elements (activities) are being performed* In the functional view, processes consist of activities that together achieve the purported goal In addition auxiliary concepts such as artefacts (products of activities) can be sued for process representation
- **Behavioural**: representing when process elements are performed, and how they are performed through feedback loops, iteration, decision making conditions, etc. process

may consist of precedence relations or information and material/information flow, with the time explicitly represented. Flow process concept focus on what happens to material and information in timeline.

- **Organizational:** where and by whom process elements are performed Process may consist of agents (performing activities) and roles (set of activities assigned to an agent). Also, process may be viewed as composed of a supplier customer partnership
- **Informational:** a perspective of the informational entities produced or manipulated by the process. In an informational perspective, processes consist of data, object, document etc.

2.6 Modelling Methodologies and Techniques

A model (Vernadat F B, 1996) is a representation of a set of components of a system or subject area that allows the system or area of concern to be understood. A system refers to a composition of interfacing or interdependent parts that work together to perform a useful function. System parts can be any combination of entities, including people, information, software, processes, equipment, products, or raw materials. A model is therefore an abstraction of a reality (or universe of discourse) expressed in terms of some formalism (or language) defined by modelling constructs for the purpose of the user Constructs of a modelling language can be defined in terms of graphical symbols, textual statements, or logic and mathematical expressions depending on the degree of formalism required Any model must clarify the system or some aspect, view or perspectives of it in order to be useful and it is important to be clear what aspect a model is supposed to express, for example a static model will display the structure of the system, whereas a dynamic model will show the activity and flow of events in the system.

The purpose of modelling is for better representation and understanding of systems. Models provide a shared vision and can be used as a common language, facilitates capitalisation of system knowledge for later reuse and provides basis for decision making concerning improvement, replacement or controlling operations and components of a systems For better representation, understanding and control of systems models are better based on aggregation of low-level information/operation and decomposition can be used whenever necessary.

Some modelling objectives are to facilitate a better human understanding and communication, and to support process improvement, process management, automated guidance in performing the process, and automated process execution (Curtis, Bill, M. I. Kellner and J. Over, 1992) In addition, they demonstrated how process modelling experimentation could be used to investigate alternatives for organizational policy formulation.

Modelling can vary widely depending on the use of the model. Some of the typical uses (Nathan and Wood 1991; Snodgrass 1993, Reimann and Sarkis1996) of modeling are summarized as

- To analyze and design the enterprise and its processes prior to implementation
- To help reduce complexity
- To communicate a common understanding of the system
- To gain stakeholder buy-in
- To act as a documentation tool for ISO 9000, TQM, Concurrent Engineering, and other efforts.

Process models are those models, which in some way describe organisations in terms of processes. Such models may be formed in a variety of ways, using a plethora of different techniques Generally, such techniques can be supported by software tools, which enable the modeller to create the models A number of specific techniques (and tools) have been developed to support the production of models. Those, which are particularly concerned with providing a representation of processes, are usually called 'process representation or process modelling techniques or methodologies An example is the IDEF family of methodologies.

A wide range of modelling techniques, are available today to facilitate a formal representation of the various aspects and perspectives of systems and subject areas. There are many modelling techniques developed and used, according to the modelling goal and perspectives The type of modelling techniques employed varies greatly and can range from Enterprise Modelling (Vernadat, 1992), through bespoke methodologies (PIM, 1993), to well-established data (Chen, 1979), process (Bravoco & Yadav, 1985) and behaviour (Murata, 1989) modelling techniques A survey conducted by Defence Information Systems Agency (SIDA) (DISA, 1995), shown that the usual tools for

Business Process Reengineering (BPR) are, functional modelling, data modelling and simulation based. Among these, the data and functional modelling tools were most widely used, whereas tools such as activity based costing (ABC), Simulation and CASE tools are less frequently used. The survey also listed those techniques which may be used most frequently in the future, these include, activity modelling, information modelling, activity based costing, interviews and surveys, and function economic analysis. For better information transfer, a formal modelling approach (Barkan et al, 1993) provides a means of achieving the required demand of representation of knowledge in a structured way and simplifies the complexity of communication One approach for achieving this is the ICAM DEFintion methods In the following sections a review of the IDEF family of methods is presented followed by descriptions of the available and widely used methods including the members of the IDEF family of methods and other modelling methodologies and techniques for process information representations and process descriptions. These include the purposes; method of representations· the concepts and syntax; and advantages and disadvantages of the techniques

2.7 IDEF Family of Modelling Methodologies Overview

IDEF (Integration DEFinition) was developed by the U S. Air Force's Integrated Computer Aided Manufacturing (ICAM) project in the late 1980's. The IDEF (ICAM Definition) method (Bravoco and Yadav, 1985a and 1985b) is a standard for requirements definition in manufacturing and other organizations. The acronym "IDEF" originally denoted "ICAM DEFinition" (Robert. P Hanrahan) now stands for "Integration DEFinition", reflecting its possible use for exchanging information between different modeling languages. A major development from (Wisnosky, Dennis E , Allen W. Batteau 1990) the ICAM program was the Integrated DEFinition methodology or IDEF as it is now called. IDEF's roots began to form when the US Air Force, in response to the identification of the need to improve manufacturing operations, established the Integrated Computer-Aided Manufacturing (ICAM) program in the mid-1970s (Mayer, Richard J., et al , 1992) As an attempt to extend the SADT method to model (CIM) enterprises IDEF was developed as part of the US Air force ICAM program in the early 1980s (ICAM, 1981). Since then, it has become the most well-known and widely used method worldwide The ICAM program identified the need for better analysis and communication techniques for people involved in improving manufacturing productivity

As a result, the ICAM program developed a series of techniques known as the IDEF (ICAM Definition) techniques, which included the following

1. IDEF0, used to produce a "function model". A function model is a structured representation of the functions, activities or processes within the modelled system or subject area.
2. IDEF1, used to produce an "information model" An information model represents the structure and semantics of information within the modelled system or subject area.
3. IDEF2, used to produce a "dynamics model" A dynamics model represents the time-varying behavioural characteristics of the modelled system or subject area

In 1983, the U S Air Force Integrated Information Support System program enhanced the IDEF1 information modeling technique to form IDEF1X (IDEF1 Extended), a semantic data modeling technique Later, IDEF has been extended to IDEF3 for enterprise behaviour modeling and IDEF5 for ontology description (Mayer et al , 1992). IDEF6 through IDEF14 have not been pursed in depth at this time These methods exist today in various stages and are intended to provide the capability to describe additional views as shown as listed in Table 2 2

Methodologies	Views
IDEF0	Function Modelling
IDEF1	Information Modelling
IDEF1X	Data Modelling
IDEF2	Simulation Model Design
IDEF3	Process Description Capture Method (Process Flow and Object State Modelling)
IDEF4	Object-Oriented Design
IDEF5	Ontology Description Capture
IDEF6	Design Rationale Capture
IDEF8	User Interface Modelling
IDEF9	Scenario-Driven IS Design
IDEF10	Implementation Architecture Modelling
IDEF11	Information Artefact Modelling
IDEF12	Organization Modelling
IDEF13	Three Schema Mapping Design
IDEF14	Network Design

Table 2 2 Suites of IDEF Methods (Current and in Development) (Mayer.Richard J. et at 1992).

As shown in table 2 1, IDEF comprises a number of modeling methodologies. Each method is useful for describing a particular perspective of a subject area or systems. The major IDEF methods (Mayer, Painter, deWitte 1992) in use are functional or activity modeling (IDEF0), information modeling (IDEF1), data modeling (IDEF1x), process description capture (IDEF3), object-oriented design (IDEF4), and ontology capture (IDEF5) Although the intention for IDEF2 was to be used as a dynamic modeling method for simulation, the numerous simulation tools commercially available have supplanted this method. Both IDEF0 and IDEF3 methodologies utilize a subordinate principle of abstraction called decomposition (Rumbaugh et al 1991), which is the breaking down of each box (activity) into more detail in a continuous manner until the greatest level of detail is achieved. (Marca and McGowan 1988)

IDEF is a rigorous methodology The reason for the rigor is to ensure a robust and complete representation. As part of this rigor, a thorough review process is used. The review cycle is enhanced by the rigid IDEF syntax. The syntax for IDEF is very explicit. However, (Larry Whitman Brian Huff, Adrien Presley, 1997) it should also be noted that there are certain characteristics of IDEF modeling that have become considered standard practice, yet they are not a strict IDEF syntactic rule The Integrated DEFinition (IDEF) methodology (Fleischer and Liker, 1997) is a suite or family of methods that supports a paradigm capable of addressing the modelling needs of an enterprise and its business areas The Integrated Definition (IDEF) is a structured analysis and design method for graphical and textual description of activities, activity relationship, information, and information relationships used to develop enterprise and system level architecture

2.8 Classification of Process Modelling Methodologies

Figure 2 2 Illustrates some of the process modelling methodologies and techniques categorised into the different perspectives of processes that they represent. The modelling methodologies, and knowledge representation languages general systems, various process perspectives: functional and behavioural, and knowledge representations were reviewed, studied and presented in the following sections The modelling techniques and knowledge representation languages are not intended to be an exhaustive list of every process representation methodologies currently available. However, they represent a sample of representations that provide some insight into different ways of representing processes and process information.

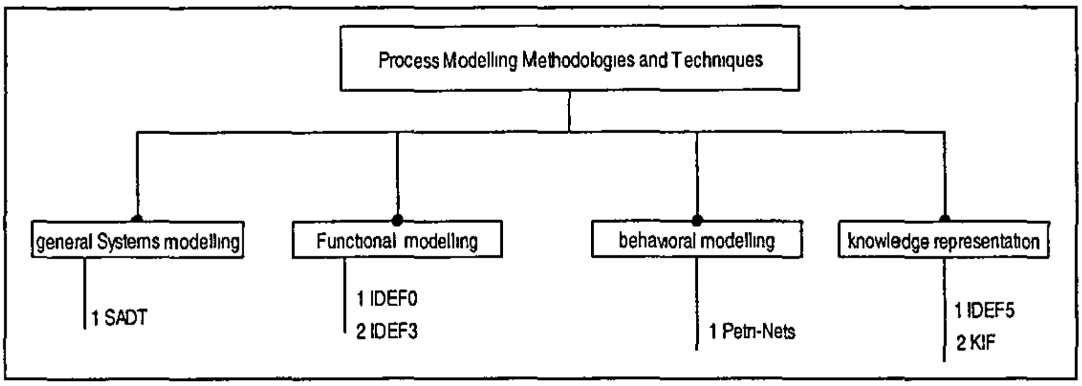


Figure 2 2 Classifications of Process Modelling Methodologies and Techniques

2.8.1 General Systems Modelling (GSM)

2.8.1.1 Structured Analysis and Design Technique (SADT)

SADT (Structured Analysis and Design Technique) (Ross, 1985) was originally developed at SofTech Inc by SofTech's Doug Ross in the early 1970s as a "system blueprinting" method for software engineering, i.e. a method for detailed requirement definition. Later on (Ross, 1997) SADT has become a full-scale methodology for problem analysis, requirement definition and functional specification applicable to many application domains As reviewed by Marca and McGowan (1988), SADT uses an unambiguous graphical notation in which a natural language is embedded. The methodology was devised to produce a specification that can be fed into other standard methodologies. A subset of SADT was the basis for the Air Force's ICAM language notation.

SADT is a graphical language with limited set of primitive constructs from which analysis and designers can compose orderly structures of any required size (Ross and Schoman, 1977). The SADT model is a set of diagrams arranged in a hierarchy. The diagram consists of a set of boxes and arrows, which represent transformation function and interfaces, respectively. The model captures multiple levels of detail in a hierarchical manner, with the higher level being a general representation of the system (Marca & McGowan, 1986). This method is similar to Data Flow Diagram (DFD) technique in that it is modular and hierarchical The flow of data is represented in both modeling techniques, but the SADT method further differentiates among input, control, and mechanism entities SADT consists (D Marca, 1998) of box and arrow diagramming language and analysis and Design technique The diagramming language is based on a simple graphical notation called the structured analysis box This construct has dual nature; the Actigram, which is used to represent activities in the case of transformation, and the datagram, which is used in representing data in the case of information or document analysis The datagrams and actigrams in SADT define a multiple views design. The analysis and design technique phasees involve the elaboration of actigrams and datagrams of the system analysed, the definition of the relationships for each type of diagram, the modification of the diagrams according to remarks and suggestions made by users and the analysis of the sequencing of activities; and detecting inconsistency in the system modeled and proposing correction for these by the creation of the model of the system desired

2.8.1.2 *Strength and Weakness of SADT*

The strength of SADT is that it is based on principle of decomposition i e. developing complex systems in to functions and sub-functions and can be applied to any kind of system However, SADT (D A. Marca and C L. McGowan, 1988) is one of the techniques that fall short on the field of customer interaction effectively encoding too early the requirements

2.8.2 IDEF0 Functional Modelling Methodology

2.8.2.1 Functional Modelling Overview

Functional aspects concern the things to be done and the way things are done in a system. Most functional modeling methods are based on functional decomposition principles. The purpose of a functional modeling approach is to describe the functionality and behavior of a system to the required level of detail for the users (Vernadat, F. B , 1996) Functionality represents actions performed in the form of functions transforming inputs into outputs over period of time. Behavior; concerns the flow of control, i e. the sequence in which things are done. Behavior governs the way functionality is performed according to occurrence of system states and real-world events.

Functionality is often associated with activities while enterprise behavior is defined by means of processes. However, few

modelling approaches make a clear distinction between activities and processes and most methods use only one recursive, construct called function or activity. The scope of functional modeling in terms of systems concerns the description of activities, either performed by human or by machines.

Formal expression: Basics of Functional Modelling

A system (Vernadat, F. B , 1996) can be made of concurrent and/or co-operative process, where each process is a flow of activities processing objects and triggered by some real-world happening or events. The activities, processes, events, and the chrematistics time and cost of activities and processes are briefly described in the following sections as follows.

(a) Activities

In essence, an activity performs something, usually by transforming its inputs into outputs it is equivalent to mathematical function. In the general case this transformation may happen only if some condition is verified. This condition can be expressed in the form of predicate (for instance expressed in first-order predicate logic) called a guard. In other words an activity transforms an input state into an output state under some condition.

Input and output states are defined by the state of system objects used or produced by the activity. The input state defines what has to be true in order to enable the activity execution. Function is a transformation, a mathematical function, a complex algorithm, a scenario to be followed by human operator, or a cognitive process such as decision making or reasoning (executed by human operator or an expert system).

Two special cases of a function can be considered as:

1. If a function is an identity function then the output state equals the input state In this case, the activity behaves like a delay (i.e. just consumes time)
2. If a function is a null function, the activity is denoted NIL and represents the activity which does nothing This is the neuter element or the set of all activities.

An activity realises a task (i.e a goal) as a partially ordered set of basic operations or action executed to perform the things to be done. Activities are performed by the functional entities and transform a input state in to an output state Activities can be classified structured or non-structured as follows,

1. Structured activity; an activity is structured if its inputs, outputs and transfer function are completely defined and if the activity behavior can be completely defined and if the activity behavior can be emulated in by a computer program A model of activities and a process plan are typical examples of structured activities
2. Non-structured; activities It is an activity for which the function is partially or not at all known, or for which even if the input vectors are known, the output is not predicable. Typical examples are human based activities such as design for which no model or formal representation exists

(b) Processes

Process is defined as a logical sequence of activities, starting with an initial set of objects and ending with a final set of objects By definition any activity is a process (i.e an elementary process made of this activity only) these NIL is a process i e that which does nothing (or neuter element of Process). More formally a process can be defined as an IDEF0 model comprising set of activities, set of objects, flow of information and material objects, set of triggering conditions or controls that need to be fulfilled before the process can start, a function such as an input and output that define the input and output objects, and flow of control between activities of the process that defines the behavior of processes. The behavior of processes can be defined for example as, sequencing, parallel and reparative activities based on flow of control between activities Hence a process can be defined as a partially ordered set of activities linked by precedence relationship, execution of which is triggered by some events and will result in some observable or quantifiable end result

Process can be classified as (Bussler, 1994) Well structured process; process for which the expected end result is known and the sequence of activities is completely defined or unstructured process; processes for which neither the end-result nor the sequence of activities are completely known.

(c) Events

Activities and processes of a system need to be synchronized. This is the role of events, which brings temporal dimension to the model. The start and end of each activity can be defined as events delimiting the duration of the activity. Events can be formally expressed as predicates or logical propositions

(d) Time and Cost

Activities and processes can be characterized by two important parameters in terms of modelling: time and cost Time is used to represent the duration of an activity or a process Similarly, a cost can be assigned to each activity for economic analysis of processes

2.8.2.2 IDEF0 Modelling Approach

IDEF0 (Integration Definition for Function Modeling) (Wisnosky & Batteau 90) was derived form a well-established graphical language, the Structured Analysis and Design Technique (SADT) and later published as IDEF0 under the ICAM (Integrated Computer Aided Manufacturing). In December 1993, the computer laboratories of the National Institute of Standards Technology (NIST) released IDEF0 as a standard for functional modelling (Federal Information Processing Standard 183). IDEF0 was the first modelling technique defined in the IDEF suite of modelling methods. It has roots in the Structured Analysis Design Technique (SADT) (Marca and McGowan 1988) (Ross 1985) developed by Ross and Softech. IDEF0 emerged from SADT and is widely used today.

IDEF0 is one of the earliest and most widely known methods for functional modelling. IDEF0 is used to model the functions of the business or its systems With IDEF0 functional modelling, you model what controls the execution of a function such as who performs the function, and what objects or data is consumed and produced by the function. Relationships can be shown between business functions, such as shared resources and dependencies. Later, Activity Based Costing may be used to determine the cost of activities based on the resources used. IDEF0 has been used extensively and therefore there are many examples of uses of this methodology in the literature. Additionally wide ranges of discussions by several authors are available in literature. For example, (Ang, et al 1999, and VanRensburg, A. & Zwemstra, N., 1995) discusses and explains how IDEF0 modeling can be used for modeling manufacturing systems. Islam

(1997) discusses the use of IDEF0 to define a generic computer integrated manufacturing system (CIM).

IDEF0 is a modeling tool used to produce a model or structured representation of the functions of a system and of the information and objects, which tie those functions together An IDEF0 model consists of diagrams and text pages describing the diagrams. Diagrams are the major components of IDEF0 model. IDEF0 methodology recognizes that successful systems development requires input and validation from the people who will ultimately use the system. The Author/Reader Cycle serves as the mechanism to facilitate communication between systems analysts and users Distributing Kits containing IDEF0 models and supporting documentation to the Reader community for comment accomplish this

IDEF0 is used to model the activities and actions of an organisation or systems. This may be done to model a wide variety of automated and non-automated systems. For new systems, IDEF0 may be used first to define the requirements and specify the functions, and then to design an implementation that meets the requirements and performs the functions For existing systems, IDEF0 can be used to analyse the functions the systems perform and to record the mechanisms (means) by which these are done The result of applying IDEF0 to a system is a model of the system that consists of a hierarchical series of diagrams, text, and glossary cross-referenced to each other IDEF0 uses input, output, control, and mechanism (ICOM) codes for graphical representation. The two primary modelling components are functions (represented on a diagram by boxes) and the data and objects that interrelate those functions (represented by arrows). The boxes represent the activities graphically and the arrows represent the information or object related to activities. The arrows generally called ICOMs, are input, output, control and mechanism depending on which side of the box the arrow is connected to. In an activity, input is transformed to output by a mechanism under a constraint or control.

2.8.2.3 Basic Concepts and Syntax of IDEF0

There are five elements in the IDEF0 functional model (Marca, D. A. and C. L McGowan 1988) as shown in **Figure 2.3.** The boxes represent functions such as activities; actions; processes or operations, and arrows represent interfaces. Boxes are denoted by an active verb phrase inside the box, such as "Perform Activity" box in **Figure 2.3** (Colquhoun, GJ, Baines, RW, Crossley, 1993,) Arrows indicate data

In IDEF0, data can be information (like "current status") or physical objects (like "raw materials"). They are named by noun phrases such as "Raw Materials" or "Tools" The position of the arrow indicates the type of information being conveyed. To describe interactions between activities, (Bravoco, R R , and Yadav S B , 1985) an ICOM (input, control, output and mechanism) presentation is used to clarify constraints and resources pertaining to an activity The arrows entering and leaving the boxes on the left and right represent "**Inputs**" and "**Outputs**", respectively. Inputs represent data needed to perform the function. Outputs show the data that is produced as a result of the function. The function transforms the inputs into the outputs. Arrows that enter from the top indicate "**Controls**", or things, which constrain or govern the function Arrows entering the bottom of the boxes are "**Mechanisms**" Mechanisms can be thought of as the person or device, which performs the function (Marca and McGowan 1988, Mayer 1992).

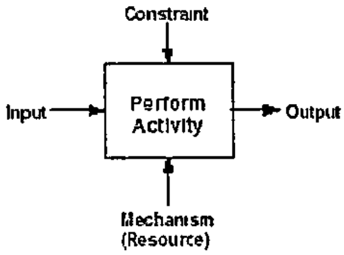


Figure 2 3 IDEF0 activity model with ICOM notation

Most of the IDEF methods utilize a subordinate principle of abstraction called decomposition (Rumbaugh, Blaha et al. 1991), which is the breaking down of each box (activity) into more detail in a continuous manner until the greatest level of detail is achieved. The IDEF0 method utilizes this principle (Marca and McGowan 1988) The method provides a diagrammatic representation of the decomposition undertaken. An example of this is shown in figure 2 4

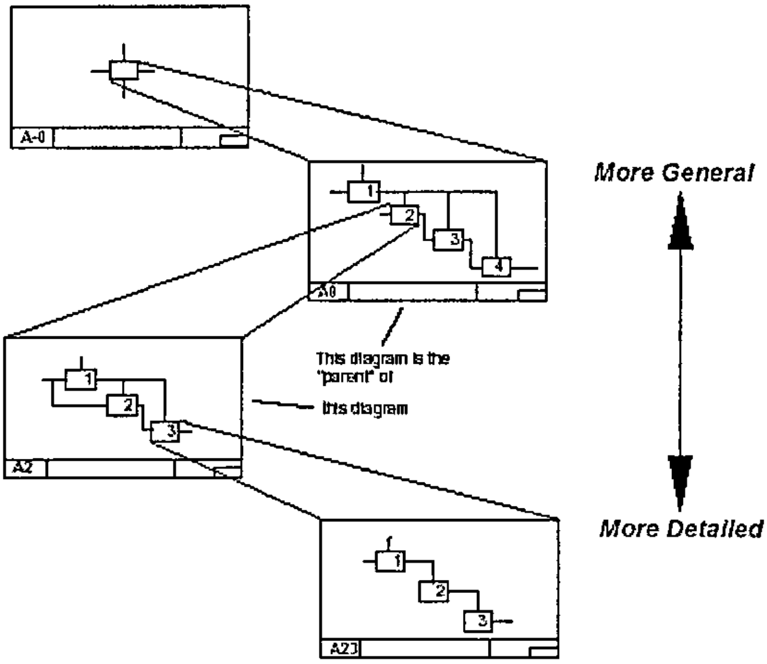


Fıgure 2 4 Decomposıtıon Overvıew (source: Rogers, Whıtman, Underdown 1998)

2.8.2.4 The IDEF0 Functional Model Objective

A functıonal model provıdes a descnptıon of the actıvıtıes performed dunng the executıon of a process (e g. desıgn, manufactunng, constructıon, operatıon, etc) The objectıve of the IDEF0 methodology ıs to decompose the process beıng analysed ınto actıvıtıes and sub-actıvıtıes ın a logıcal and progressıve manner. The decomposıtıon of actıvıtıes ıs hıerarchıcal where the top of the model ıs less detaıled than those at the bottom. As a functıon modellıng language, IDEF0 has the followıng charactenstics:

- ❑ It ıs comprehensıve and expressıve, capable of graphıcally representıng a wıde vanety of busıness, manufactunng and other types of enterpnse operatıons to any level of detaıl.
- ❑ It ıs a coherent and sımple language, provıdıng for ngorous and precıse expressıon, and promoting consıstency of usage and ınterpretatıon.
- ❑ It enhances communıcatıon between systems analysts, developers and users through ease of learnıng and ıts emphasıs on hıerarchıcal exposıtıon of detaıl.
- ❑ It ıs well tested and proven, through many years of use in Aır Force and other government development projects, and by pnvate ındustry.

- It can be generated by a variety of computer graphics tools; numerous commercial products specifically support development and analysis of IDEF0 diagrams and models

2.8.2.5 Strengths and Weaknesses of IDEF0

As described in an analysis of the strengths and weaknesses of IDEF0 by Knowledge Based Systems, Incorporated. "The primary strength of IDEF0 is that the method has proven effective in detailing the system activities for function modeling, the original structured analysis communication goal for IDEF0 Additionally, the description of the activities of a system can be easily refined into greater and greater detail until the model is as descriptive as necessary for the decision-making task at hand. In fact, one of the observed problems with IDEF0 models is that they often are so concise that they are understandable only if the reader is a domain expert or has participated in the model development "

2.8.3 IDEF3 "Process Flow and Object State Description Capture" Method

2.8.3.1 Overview

IDEF3 Process Flow and Object State Description Capture Method is another member of the IDEF family of methods, which were developed under the Information Integration for Concurrent Engineering (IICE) project funded by the U.S. Air Force (Mayer et al. 1992) Knowledge base systems, Inc., which is the prime contractor for IICE and developer of IDEF3. IDEF3 is structured method designed to capture the behavioral aspects of an existing or proposed system. It is an expressively powerful language for process information capture and representations. It enables knowledge acquisition by direct capture of assertions about processes and events including objects that participate in the process, assertions about supporting objects, and the precedence (activity sequence), and causality relationships between processes and events in the environment. The resulting descriptions that are captured by the IDEF3 can then be used to facilitate the construction of analytical and design models.

2.8.3.2 Process Descriptions and Models

IDEF3 is a process description method not a modelling language. Mayer argued that it is important to distinguish between models and descriptions (Mayer et al. 1988). It was emphasised that although models may well be constructed from descriptions, IDEF3 is not for a method construction of models but the formal representation of descriptions and the information they convey. The distinction between models and description is that a "model" can be characterized as an idealized system of objects, properties and relationships that is designed to imitate in certain relevant respects the character of a given real world systems. The power of a model comes from its ability to simplify the real world system it represents, and to predict certain facts about that system in virtue of corresponding facts within the model (Corynen, 1975). A model is thus itself, in a certain sense, a complete system. In order to be an acceptable model of a given or imagined real world situation, it must satisfy certain "axioms" or conditions derived from the real world system. On the other hand, a "description" is a recording of facts or beliefs about the world around us. Such description are generally partial, some facts or beliefs can be omitted from a description that do not seem to be relevant or forgotten in the courses of describing a system or a real world situation. There are no preconditions on an acceptable description, no axioms to be satisfied, short of simple accuracy as far as it goes; descriptions, it might be said, that are assumed to be true, but incomplete. One very powerful use of models is to fill in the gaps in descriptions (Menzel, et al 1990). The accumulation of descriptions is thus prior to and distinct from the construction of models. The conditions that one may put on acceptable models are derived from descriptions (although not necessarily IDEF3 descriptions) of a domain or a system that is the data from which the model is constructed. Hence descriptions are basis to model building process. An accurate treatment of such descriptions requires two components: the syntactic and the semantics. That is, it is important that there is an effective means of representing the descriptions, a means of capturing their logical form. IDEF3 makes use of the standard language of first-order logic as its formal base that permits a rich and flexible means of expressing the logical forms of any descriptive statement. On the other hand, there must also be a rigorous account of their information content. A variant of first-order semantics can be used to represent information crucial/intrinsic to process description. This approach enables interpretation of the intended meaning of a given description in terms of a semantic structure that corresponds in a natural way to the real world situation being described.

2.8.3.3 Basic Concepts of IDEF3 Technique

IDEF3 (Mayer, et al 1995) is essentially a diagramming/ graphical language, supported by a set of forms to collect requirements for capturing information about the processes and participant objects involved in a system. The IDEF3 methodology (Mayer, Cullinane et al 1992) is capable of providing different users views of temporal precedence and causality relationships via two main knowledge acquisition or description modes. One, the Process Flow Description (PFD), provides a process-centered view of a system, while the other, the Object State Transition Network (OSTN), allows an object-centered view. Both of these complementary representations employ the same basic diagrammatic notation scheme, featuring series of boxes (either square or oblong), circles, and interconnecting arcs Textual elaboration forms can also be attached to each of the graphical icons, to provide additional information The meanings and usage (i e , semantics and syntax) of these graphical entities is dependent upon which of the two description modes is being viewed. The information is presented in process flow diagrams, object state transition networks and elaboration diagrams. The resulting diagrams and text comprise what is termed a " Process Descriptions" as opposed to the focus of what is produced by other IDEF methods whose product is a " Process Model".

2.8.3.4 IDEF3 Representation of Process Descriptions

The process flow description (PFD) provides a process-centred view of a process by capturing description of processes and the network of relations that exists between processes within the context of the overall scenario in which they occur. Process is a set of activities, decision, action, function or any types of happenings that stand in a certain relationship to one another and to objects over a time point (Lee et al.96). The Object state transition network (OSTN); provides a object-centred view of a process, by capturing information about how objects of various kinds are transformed into other kinds and given kinds change in states through a process and context setting information about important relations between objects in a process. Objects are the entities manipulated by the processes Objects are any physical or conceptual things that is recognized and referred to, by participants in the domain, as part of their descriptions of what happens in their domain. IDEF3 defines an object as an abstraction of a real-world entity, which intervenes in a process description.

The Process Flow description (PFD) and the Object State Transition Network (OSTN) use the basic elements of the IDEF3 language to capture and express the assertions that form the description. Graphical projections of the information contained in process descriptions are created using IDEF3's graphical language. These graphical projections used to both record process information directly and as a mechanism to display process information are called schematics. Two types of IDEF3 schematics are parallel to the two process knowledge acquisition strategies The IDEF3 Process Schematic displays a process-centred view of a scenario. Object Schematics support the graphical display of object-centred information Object Schematics that display an object-centred view of a single scenario are called Transition Schematics. Transition Schematics that display additional objects and object relations to provide context-setting information are called Enhanced Transition Schematics. Object Schematics that display object-centred information spanning multiple scenarios are simply called Object Schematics. The scenario concept is used to organize both the process-centred and object-centred views. The collection of scenarios and the information they serve to organize is the IDEF3 Process Description.

2.8.3.5 Scenarios: The Organizing Structure for IDEF3 Process Descriptions

The notion of a scenario (Mayer, et al 1995) is used as the basic organizing structure for IDEF3 Process Descriptions. A scenario can be thought of as a recurring situation, a set of situations that describe a typical class of problems addressed by an organization or system, or the setting within which a process occurs Scenarios establish the focus and boundary conditions of a description. Using scenarios in this way exploits the tendency of humans to describe what they know in terms of an ordered sequence of activities within the context of a given scenario or situation Scenarios also provide a convenient vehicle to organize collections of process-centered knowledge.

2.8.3.6 Description Representation Concepts and Syntax of IDEF3 Schematics

IDEF3 uses a rigid syntax that eliminates model ambiguity. The basic elemental notations of IDEF3 Process Descriptions Capture Method consists of a series of square and oblong boxes, circles and arcs, which link them, attached to each icon is an elaboration form, which contains a description of that icon, reference label etc, and details of related objects, facts and constraints, acting upon it. In the following sections the process description representation concepts and syntax available in the two types of IDEF3 schematics is described briefly.

The Process Schematics: Process Centred Views (PFD)

A process flow description (PFD). captures knowledge of "how things work" in an organization, e g , the description of what happens to a part as it flows through a sequence of manufacturing processes and provides a process centred view IDEF3 Process Schematics are the primary means for capturing, managing, and displaying process-centered knowledge These schematics provide a graphical medium for communicating knowledge about processes This includes knowledge about events and activities, the objects that participate in those occurrences, and the constraining relations that govern the behavior of occurrences.

A process-centered description can be constructed systematically, using the basic building blocks of the IDEF3 schematic language, linked together in different ways. These building blocks have specific semantics associated with them. That is, they are used to represent certain kinds of activities or relationships in a real world The IDEF3 diagrams shown in **appendix B** depicts Process Schematics of an architectural design, pre-tender cost estimating and scheduling processes developed in this project. Textual elaboration forms are also attached to each of the graphical icons, providing additional information as shown in **Appendix C.** In IDEF3, scenarios bound the context of descriptions and are convenient artifacts for describing similar situations from different perspectives.

The development of an IDEF3 process flow description (Process Schematics) consists of expressing facts in terms of the IDEF3 basic description building blocks or syntax Process schematics tend to be the most familiar and broadly used component of theIDEF3 method These schematics provide a visualisation mechanism for process-centred descriptions of a scenario in terms of the basic graphical building blocks. The graphical elements that comprise process schematics include Unit of Behaviour (UOB) boxes, links, junctions, referents, and notes. Referents and notes are constructs that are common across process and object schematics. The IDEF3 method allows users to capture descriptions at varying levels of abstractions by providing a mechanism called decomposition Decomposition provides a means of organizing a more detailed description of UOB representing an activity. The decomposition schematic follows the same syntactic rules as those for a scenario and is created using the same IDEF3 elements. The decomposition of the same UOB more than one can be employed for representing different points of view or providing greater details of the processing relating to the activity or UoB

A PFD displays a sequence of Units of Behaviour (UoB), which represent activities, actions, processes or operations. These are linked together by precedence arcs or Links that connect the boxes and describe the relationship between the various UOBs. Junctions explicitly describe the logic of multiple links either coming together or spreading apart or the branching of processes. In addition the junctions can show if the processes are being carried out in synchronization or asynchronisation This notation may impose timing constraints on the process flow.

The Object Schematics: Object-Centered Views (OSTN)

Object state transition network (OSTN). The object state transition network description summarizes the allowable transitions an object may undergo throughout a particular process.

IDEF3 Object Schematics capture, manage, and display object-centered descriptions of a process that is, information about how objects of various kinds are transformed into other kinds of things through a process, how objects of a given kind change states through a process, or context-setting information about important relations among objects in a process Object Schematics may be developed in the context of a single scenario, thus characterizing the state transitions traversed by participating objects in an occurrence of the scenario These Transition Schematics allow users to specify the rules that govern the transitions between object states in a scenario occurrence. Alternatively, Object Schematics may evolve in a more opportunistic fashion, capturing descriptions of objects, object states, and their transitions across multiple scenarios. Object Schematics developed in this fashion make no attempt to define the structure for object state change behavior in a scenario occurrence This cross-scenario Object Schematic development approach is often useful when exploring what object-centered process information merits a more detailed focus or when attempting to discover context setting information about the objects encountered in a description. IDEF3 Object Schematics are developed to provide an object-centered description of a particular process or scenario. The IDEF3 Object Schematics diagrams shown in **appendix B** depicts an object-centered view of the process schematics developed for the architectural design, pre-tender cost estimating and scheduling processes developed in this project. Textual elaboration forms are also attached to each of the graphical icons, providing additional information as shown in **Appendix C.** Transition Schematics therefore tend to dominate the attention of those developing IDEF3 Object Schematics.

Object state transition network (OSTN) diagrams capture object-centred views of processes, which cut across the process diagrams and summarize the allowable transitions The basic elements are nodes circles and arcs (arrows) The nodes in the diagram indicate the different states of the object. The arcs represent the transition that objects can make. Object states and state transition arcs are the key elements of an OSTN diagram A detailed specification of the IDEF3 constructs can be fond in IDEF3 Process Description Capture Method Report (Mayer, et al 1995)

2.8.3.7 Strengths and Weaknesses of IDEF3

Among the existing process representation methodologies, the IDEF3 process description method best meets the requirement for representing process characteristics in a domain (Knutilla, A., et. al , 1998). It provides the best coverage of all requirements necessary for describing processes from process and participating objects point of view The methodology stops short of providing the ability to construct predictive simulation-based models; rather, it is a method to obtain structured descriptions of what a system actually can or will do in practice.

2.8.4 Knowledge Representations

What is Knowledge Representation?

A knowledge representation (KR) is best understood in terms of the five fundamental roles that it plays as described (R. Davis, H Shrobe, and P. Szolovits, 1993):

1. KR is a Surrogate: Any intelligent entity that wishes to reason about its world encounters an important, in-escapable fact: reasoning is a process that goes on internally, while most things it wishes to reason about exist only externally
2. KR is a Set of Ontological Commitments· If, as we have argued, all representations are imperfect approximations to reality, each approximation attending to some things and ignoring others, then in selecting any representation we are in the very same act unavoidably making a set of decisions about how and what to see in the world. That is, selecting a representation means making a set of ontological commitments.
3. KR is a Fragmentary Theory of Intelligent Reasoning: The third role for a representation is as a fragmentary theory of intelligent reasoning This role comes about because the initial conception of a representation is typically motivated by some insight indicating how people reason intelligently, or by some belief about what it means to reason intelligently at all.
4. KR is a Medium for Efficient Computation: From a purely mechanistic view, reasoning in machines (and somewhat more debatably, in people) is a computational process. Simply put, to use a representation we must compute with it. As a result, questions about computational efficiency are inevitably central to the notion of representation
5. KR is a Medium of Human Expression: Finally, knowledge representations are also the means by which we express things about the world, the medium of expression and communication in which we tell the machine (and perhaps one another) about the world. This role for representations is inevitable so long as we need to tell the machine (or other people) about the world, and so long as we do so by creating and communicating representations.

2.8.4.1 IDEF5 Ontology Description Capture Method

Overview

Historically, ontologies arose from the branch of philosophy known as metaphysics, which deals with the nature of reality of what exists. The traditional goal of ontological inquiry, in particular, is to divide the world "at its joints": to discover those fundamental categories or kinds that define the objects of the world. According to Benjamin, et al

(1995), "an ontology is a description of the kinds of things, both physical and conceptual, that make up a given domain, their associated properties, and the relationships that hold among them as represented by the terminology in that domain." Within the IDEF suite, the IDEF5 ontology capture method (Benjamin et al., 1995) was developed for this purpose. IDEF5 Standardized procedures enable the ability to represent ontology information in an intuitive and natural form

IDEF5 (Mayer et al. 1992) is one of the IDEF (ICAM DEFinition,) methods developed under the Information Integration for Concurrent Engineering (IICE) project funded by the U.S. Air Force Knowledge base systems, Inc , for enterprise knowledge representation The IDEF5 ontology description capture method (KBSI 1994c) provides a theoretically and empirically well-grounded method specifically designed to assist in creating, modifying, and maintaining ontologies IDEF5 proposes a methodology to create domain ontology and provides a language to represent concepts (meta-ontology). IDEF5 meta-ontology contains the meta-concepts process and state This is the only difference that characterises IDEF among other languages like KIF and Ontolingua, used to represent domain ontologies These dynamic concepts make IDEF5 an example of meta-ontology for representing enterprise ontologies

Basic Concepts of IDEF5

Knowledge acquisition with IDEF5 is enabled by the direct capture of assertions about real-world objects and their interrelationships in an intuitive and natural form. IDEF5 proposes a set of meta-concepts for ontology. They are: kinds, characterized by certain properties and attributes, and process; state of an individual of a certain kind that can be modified by a process; relations among kinds; the decomposition relation; the sub-kind relation (a sort of refinement) Hence, there is the possibility of building second order properties and relations This first set of concept is the base for the schematic languages of the IDEF family, like IDEF3 ((Mayer et al.92) for diagrammatic representation processes The method also provides facilities for diagrammatic representations of ontology and a structured text language for detailed ontology characterization is present.

IDEF5 Ontology Languages

To support the ontology development process IDEF5 proposes two languages: the schematic language and the elaboration language (Knowledge Based Systems, Inc.

(KBSI, 2000). The schematic language is a graphical language, specifically tailored to enable domain experts to express the most common forms of ontological information. This enables average users both to input the basic information needed for a first-cut ontology and to augment or revise existing ontologies with new information. The elaboration language is a structured textual language that allows detailed characterization of the elements in the ontology. Employing these two developments supports, three kinds of expressions can be formulated: definition, term and sentence The elaboration language uses KIF (Knowledge Interchange Format) http //logic stanford.edu/kif/kif.html as its foundation.

The purpose of these schematics, like that of any representation, is to represent information visually. Thus, semantic rules must be provided for interpreting every possible schematic. These rules are provided by outlining the rules for interpreting the most basic constructs of the language, then applying them recursively to more complex constructs. This task falls to the Elaboration Language The Schematic Language is, however, useful for constructing first-cut ontologies in which the central concern is to record, in a briefly, the basic elements that exist in a domain, their characteristic properties, and the salient relationships that can be obtained among objects of those kinds and among the kinds themselves There are four primary schematic types derived from the basic IDEF5 Schematic language, which can be used to capture ontology information directly in a form that is intuitive to the domain experts. These are: Classification Schematics, Composition Schematics, Relation Schematics, and Object State Schematics. A detailed explanation of the IDEF5 method can be found in KBSI (1994c).

IDEF5 Primary Schematic Types

Classification schematics: provide mechanisms for the organization of knowledge into logical taxonomies The description subsumption and natural kind classification are the two types of classification schematics.

Composition schematic: serves as mechanisms to represent graphically the "part-of" relation that is so common among components of ontology. In particular, this capability enables users to express facts about the composition of a given kind of object

Relation schematics· allow ontology developers to visualize and understand relations among kinds in a domain, and can also be used to capture and display relations between first-order relations In other process modelling/representations methodologies there is no clean division between information about kinds, states and processes.

Object State Schematics: The IDEF5 schematic language enables modellers to express fairly detailed object-centred process information (i e., information about kinds of objects and the various states they can be in relative to certain processes).

Strengths and Weakness of IDEF5 Method

IDEF5 is also intended to provide a mechanism for reference models to facilitate reorganization of similarities between components of a system as well as with external systems. The weakness of the method is that the definitions of kinds in IDEF5 are not rigorous. IDEF5 does not impose formal constraints to instances that are required to share all the properties, i.e. properties that are individually necessary and jointly sufficient for an individual being in a kind. Consequently also the sub-kind relation cannot be defined in rigorous manner.

2.8.4.2 Knowledge Interchange Format (KIF)

Overview

KIF (Genesereth & Fikes et al , 1992) is a computer-oriented language aimed atknowledge sharing among disparate programs developed by the Interlingua working group of the DARPA Knowledge Sharing Effort. At the initial stage of design, KIF needed to have a formally defined declarative semantics, sufficient expressive power, and a structure that enables semi-automatic translation into and out of conventional knowledge representation language. Furthermore, KIF was needed to decrease the number of translators per knowledge base to one· local language to KIF and back. KIF has a tree-like, structured syntax as well as a corresponding linear, ASCII, list-based syntax. Intuitively, KIF terms denote objects in the universe of discourse, and every sentence is either true or false. KIF provides the user with a set of basic objects, which are described by its standard vocabulary on numbers, lists, sets, functions, and relations Besides the basic objects, KIF also allows expressions of meta-linguistic knowledge as well as provides supports for representations necessary for non-monotonic reasoning KIF is also not intended as an internal representation for knowledge within computer systems or within closely related sets of computer systems (though the language can be used for this purpose as well).

Typically, when a computer system reads a knowledge base in KIF, it converts the data into its own internal form (specialized pointer structures, arrays, etc.). All computation is done using these internal forms. When the computer system needs to communicate with another computer system, it maps its internal data structures into KIF.

The purpose of KIF is roughly analogous to that of Postscript. Postscript is commonly used by text and graphics formatting systems in communicating information about documents to printers While KIF is not as efficient as a specialized representation for knowledge or as perspicuous as a specialized display (when printed in its list form), it is a programmer-readable language and thereby facilitates the independent development of knowledge-manipulation programs.

The following categorical features are essential to the design of KIF

- The language has declarative semantics. It is possible to understand the meaning of expressions in the language without appeal to an interpreter for manipulating those expressions In this way, KIF differs from other languages that are based on specific interpreters, such as Prolog.
- The language is logically comprehensive -- it provides for the expression of arbitrary sentences in predicate calculus. In this way, it differs from relational database languages (many of which are confined to ground atomic sentences) and Prolog-like languages (that are confined to Horn clauses).
- The language provides for the representation of knowledge about the representation of knowledge. This allows us to make all knowledge representation decisions explicit and permits us to introduce new knowledge representation constructs without changing the language

In addition to these essential features, KIF is designed to maximize the following additional features (to the extent possible while preserving the preceding features):

- Translatability: A central operational requirement for KIF is that it enable practical means of translating declarative knowledge bases to and from typical knowledge representation languages.
- Readability: Although KIF is not intended primarily as a language for interaction with humans, human readability facilitates its use in describing representation

language semantics, its use as a publication language for example knowledge bases, its use in assisting humans with knowledge base translation problems, etc

- Implementability Although KIF is not intended for use within programs as a representation or communication language, it can be used for that purpose if so desired.

KIF Syntax

The syntax of KIF is most easily described in three layers. First, there are the basic characters of the language. These characters can be combined to form lexemes Finally, the lexemes of the language can be combined to form grammatically legal expressions. Although this layering is not strictly essential to the specification of KIF, it simplifies the description of the syntax by dealing with white space at the lexeme level and eliminating that detail from the expression level. The syntax of KIF is presented using a modified BNF notation. All nonterminals and BNF punctuation are written in boldface, while characters in KIF are expressed in plain font.

The notation {x1, . ,xn} means the set of terminals x1,...,xn. The notation **[nonterminal]** means zero or one instances of **nonterminal**; **nonterminal*** means zero or more occurrences; **nonterminal+** means one or more occurrences; **nonterminal^n** means **n** occurrences. The notation **nonterminal1 - nonterminal2** refers to all of the members of **nonterminal1** except for those in **nonterminal2**. The notation **int(n)** denotes the decimal representation of integer n. The nonterminals **space**, **tab**, **return**, **linefeed**, and **page** refer to the characters corresponding to ascii codes 32, 9, 13, 10, and 12, respectively. The nonterminal **character** denotes the set of all 128 ascii characters The nonterminal **empty** denotes the empty string

Characters

The alphabet of KIF consists of 7 bit blocks of data. In this document, we refer to KIF data blocks via their usual ASCII encoding as characters (as given in ISO 646·1983). KIF characters are classified as upper case letters, lower case letters, digits, alpha characters (non-alphabetic characters that are used in the same way that letters are used), special characters, white space, and other characters (every ascii character that is not in one of the other categories)

upper ::= A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

lower ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z

digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

alpha ::= ! | $ | % | & | * | + | - | | / | < | = | > | ? | @ | _ | ~ |

special ::= " | # | ' | (|) | , | \ | ^ | `

white ::= space | tab | return | linefeed | page

A normal character is either an upper case character; a lower case character; a digit; or an alpha character

normal ::= upper | lower | digit | alpha

Lexemes

The process of converting characters into lexemes in called lexical analysis The input to this process is a stream of characters, and the output is a stream of lexemes. The function of a lexical analyzer is cyclic. It reads characters from the input string until it encounters a character that cannot be combined with previous characters to form a legal lexeme. When this happens, it outputs the lexeme corresponding to the previously read characters. It then starts the process over again with the new character. White space causes a break in the lexical analysis process but otherwise is discarded.

There are five types of lexemes in KIF, special lexemes, words, character references, character strings, and character blocks. Each special character forms its own lexeme It cannot be combined with other characters to form more complex lexemes, except through the ``escape" syntax described below. A word is a contiguous sequence of (1) normal characters or (2) other characters preceded by the escape character \.

word ::= normal | word normal | word\character

A character reference consists of the characters #, \, and any character. Character references allow us to refer to characters as characters and differentiate them from one-character symbols, which may refer to other objects.

charref ::= #\character

A character string is a series of characters enclosed in quotation marks. The escape character \ is used to permit the inclusion of quotation marks and the \ character itself within such strings

string ::= "quotable"

quotable ::= empty | quotable strchar | quotable\character

strchar ::= character - {",\}

Sometimes it is desirable to Character blocks permit the group together a sequence of arbitrary bits or characters without imposing escape characters, e g to encode images, audio, or video in special formats through the use of a prefix that specifies how many of the following characters are grouped together in this way A character block consists of the character # followed by the decimal encoding of a positive integer **n**, the character q or Q, and then **n** arbitrary characters

block ::= # int(n) q **character^n | # int(n)** Q **character^n**

For the purpose of grammatical analysis the class of words is subdivide further, viz. as variables, operators, and constants.

A variable is a word in which the first character is ? or @ A variable that begins with ? is called an individual variable. A variable that begins with an @ is called a sequence variable

variable ::= indvar | seqvar

indvar ::= ?word

seqvar ::= @word

Operators are used in forming complex expressions of various sorts There are three types of operators in KIF term operators, sentence operators, and definition operators. Term operators are used in forming complex terms. Sentence operators and user operators are used in forming complex sentences. Definition operators are used in forming definitions

operator ::= termop | sentop | defop

termop ::= value | listof | quote | if

sentop ::= holds | = | /= | not | and | or | => | <= | <=> | forall | exists

defop ::= defobject | defunction | defrelation | deflogical | := | :-> | :<= | :=>

All other words are called constants.

constant ::= word - variable – operator

Semantically, there are four categories of constants in KIF: object constants, function constants, relation constants, and logical constants. Object constants are used to denote individual objects. Function constants denote functions on those objects. Relation constants denote relations. Logical constants express conditions about the world and are either true or false. KIF is unusual among logical languages in that there is no syntactic distinction among these four types of constants, any constant can be used where any other constant can be used The differences between these categories of constants are entirely semantic.

Expressions

The legal expressions of KIF are formed from lexemes according to the rules presented in this section There are three disjoint types of expressions in the language· terms, sentences, and definitions. Terms are used to denote objects in the world being described, sentences are used to express facts about the world, and definitions are used to define constants. Definitions and sentences are called forms A knowledge base is a finite set of forms.

There are nine types of terms in KIF -- individual variables, constants, character references, character strings, character blocks, functional terms, list terms, quotations, and logical terms. Individual variables, constants, character references, strings, and blocks were discussed earlier.

term ::= indvar | constant | charref | string | block |funterm | listterm | quoterm | logterm

An implicit functional term consists of a constant and an arbitrary number of argument terms, terminated by an optional sequence variable and surrounded by matching parentheses. Note that there is no syntactic restriction on the number of argument terms; arity restrictions in KIF are treated semantically

funterm ::= (constant term* [seqvar])

A explicit functional term consists of the operator value and one or more argument terms, terminated by an optional sequence variable and surrounded by matching parentheses.

funterm ::= (value **term term* [seqvar])**

A list term consists of the listof operator and a finite list of terms, terminated by an optional sequence variable and enclosed in matching parentheses.

listterm ::= (listof **term* [seqvar])**

Quotations involve the quote operator and an arbitrary list expression. A list expression is either an atom or a sequence of list expressions surrounded by parentheses. An atom is either a word or a character reference or a character string or a character block Note that the list expression embedded within a quotation need not be a legal expression in KIF

quoterm ::= (quote **listexpr) | 'listexpr**

listexpr ::= atom | (listexpr*)

atom ::= word | charref | string | block

Logical terms involve the if and cond operators. The if form allows for the testing of a single condition or multiple conditions. An optional term at the end allows for the specification of a default value when all of the conditions are false. The cond form is similar but groups the pairs of sentences and terms within parentheses and has no optional term at the end.

logterm ::= (if **logpair+ [term]**)

logpair ::= sentence term

logterm ::= (cond **logitem***)

logitem ::= (sentence term)

The following BNF defines the set of legal sentences in KIF There are six types of sentences We have already mentioned logical constants.

sentence ::= constant | equation | inequality | relsent | logsent | quantsent

An equation consists of the = operator and two terms. An inequality consist of the /= operator and two terms

equation ::= (= **term term**)

inequality ::= (/= term term)

An implicit relational sentence consists of a constant and an arbitrary number of argument terms, terminated by an optional sequence variable. As with functional terms, there is no syntactic restriction on the number of argument terms in a relation sentence.

relsent ::= (constant term* [seqvar])

An explicit relational sentence consists of the operator holds and one or more argument terms, terminated by an optional sequence variable and surrounded by matching parentheses

```
relsent ::= (holds term term* [seqvar])
```

It is noteworthy that the syntax of implicit relational sentences is the same as that of implicit functional terms On the other hand, their meanings are different. Fortunately, the context of each such expression determines its type (as an embedded term in one case or as a top-level sentence or argument to some sentential operator in the other case); and so this slight ambiguity causes no problems.

The syntax of logical sentences depends on the logical operator involved. A sentence involving the not operator is called a negation. A sentence involving the and operator is called a conjunction, and the arguments are called conjuncts A sentence involving the or operator is called a disjunction, and the arguments are called disjuncts A sentence involving the => operator is called an implication, all of its arguments but the last are called antecedents; and the last argument is called the consequent. A sentence involving the <= operator is called a reverse implication, its first argument is called the consequent; and the remaining arguments are called the antecedents. A sentence involving the <=> operator is called an equivalence.

```
logsent ::= (not sentence) |
            (and sentence*) |
            (or sentence*) |
            (=> sentence* sentence) |
            (<= sentence sentence*) |
            (<=> sentence sentence)
```

There are two types of quantified sentences: a universally quantified sentence is signalled by the use of the "forall" operator, and an existentially quantified sentence is signalled by the use of the "exists" operator. The first argument in each case is a list of variable specifications. A variable specification is either a variable or a list consisting of a variable and a term denoting a relation that restricts the domain of the specified variable

```
quantsent ::= (forall (varspec+) sentence) |
              (exists (varspec+) sentence)
 varspec ::= variable | (variable constant)
```

Note that, according to these rules, it is permissible to write sentences with free variables, i e. variables that do not occur within the scope of any enclosing quantifiers The significance of the free variables in a sentence depends on the use of the sentence When we assert the truth of a sentence with free variables, we are, in effect, saying that the sentence is true for all values of the free variables, i e. the variables are universally quantified. When we ask whether a sentence with free variables is true, we are, in effect, asking whether there are any values for the free variables for which the sentence is true, i e. the variables are existentially quantified.

The following BNF defines the set of legal KIF definitions There are three types of definitions: unrestricted, complete, and partial. Within each type, there are four cases, one for each category of constant. Object constants are defined using the "defobject" operator Function constants are defined using the "deffunction" operator. Relation constants are defined using the "defrelation" operator Logical constants are defined using the "deflogical" operator

```
definition ::= unrestricted | complete | partial
unrestricted ::=
    (defobject constant [string] sentence*) |
    (deffunction constant [string] sentence*) |
    (defrelation constant [string] sentence*) |
    (deflogical constant [string] sentence*)

complete ::=
    (defobject constant [string] := term) |
    (deffunction constant (indvar* [seqvar]) [string] := term) |
    (defrelation constant (indvar* [seqvar]) [string] := sentence) |
    (deflogical constant [string] := sentence)

partial ::=
    (defobject constant [string] :-> indvar :<= sentence) |
    (defobject constant [string] :-> indvar :=> sentence) |
    (deffunction constant (indvar* [seqvar]) [string] :-> indvar :<= sentence) |
    (deffunction constant (indvar* [seqvar]) [string]:-> indvar :=> sentence) |
    (defrelation constant (indvar* [seqvar]) [string] :<= sentence) |
    (defrelation constant (indvar* [seqvar]) [string]:=> sentence) |
    (deflogical constant [string] :<= sentence)
    (deflogical constant [string] :=> sentence)
```

A form in KIF is either a sentence or a definition.

form ::= sentence | definition

It is important to note that definitions are top-level constructs While definitions contain sentences, they are not themselves sentences and, therefore, cannot be written as constituent parts of sentences or other definitions (unless they occur inside of a quotation. A knowledge base is a finite set of forms. It is important to keep in mind that a knowledge base is a set of sentences, not a sequence; and, therefore, the order of forms within a knowledge base is unimportant. Order may have heuristic value to deductive programs by suggesting an order in which to use those sentences; however, this implicit approach to knowledge exchange lies outside of the definition of KIF.

2.8.5 Behavioural Modelling

2.8.5.1 Petri Nets

A Petri Net (Reisig 1992, Peterson 1981) is a graphical language that is appropriate for modeling systems with concurrency. Petri Nets has been under development since the beginning of the 60's, when Carl Adam Petri defined the language in his PhD thesis (Kommunikation mit Automaten). The language was created to represent a net-like, mathematical tool for the study of communication with automata such that the concept of concurrently occurring events could be expressed. Petri Nets is a well-suited technique for modeling and analysis of concurrency, synchronization and conflict, communication protocols, performance evaluation, fault tolerant systems, and manufacturing control systems and hence are a natural choice for modeling complex systems (Viswanadham and Narahari, 1992). Petri Nets support advanced and formal analyses such as the reachability of various process states, detection of deadlocks, and behavioural ambiguities. According to Curtis et al. (1992) such analyses are critical to permit evaluation of a proposed process for syntactic correctness, completeness, consistency, risks, and opportunities for improvements

Since it was first introduced, Petri Nets have been modified and extended by various researchers to allow for more powerful modeling capabilities: (David and Alla, 1991; Desrochers and Al-Jaar, 1994, Kamath and Viswanadham, 1986; Zurawski and Zhou, 1994). Some of the important extensions include (i) Stochastic Petri nets, which capture timed activities/events by using (probabilistic) timed transitions these Petri net models can then be translated into a Markov chain for further analysis, and (ii) Colored Petri nets, where the tokens can be colored (or carry attribute information) to differentiate the entities that flow through the system The other variations include Timed Petri Nets, Predicate/Transition Nets, Object Petri Nets, Compact Petri Nets, Role-based Extended Petri Net Models, Hierarchical Petri Nets, and Queueing Petri Nets

Strength and Weaknesses of Petri Nets

Some of the arguments in favor of using Petri nets as formal representations of business processes are.

- They represent a hierarchical modeling tool with a well-developed mathematical foundation (Murata, 1989)
- Petri nets support both qualitative and quantitative analyses and many computerized tools are available for the same (Desrochers and Al-Jaar 1994).
- They lend themselves very well to simulation/execution on distributed/parallel architectures which is essential for representations of realistically-sized enterprise systems (Ferscha, 1998)
- They have been used by themselves, or in conjunction with other techniques to model business processes (Sagoo and Boardman, 1998; van der Aalst, 1999a,b).

2.8.6 Data Flow Diagrams (DFD)

Data flow Diagrams (DFDs) (Grady 1993, Scotti 1994) is another staple of the system engineer's toolbox, and have also found extensive usage in many other fields including software and information design. It is primarily used as a tool for performing structured systems analyses to explore the relationships between processes and the data that they transform or create DFDs represent the flow of data throughout a process or between processes, depicting a system from a data perspective (of those who use the data), as opposed to a control perspective (of those who act upon the data), or a resource perspective (what is needed by whom and why, to do what).

DFDs is a graphical/diagrammatic representation that hierarchically decomposable, each DFD at a given level can be seen to "explodes" a process from a preceding level. DFDs include specifications of the boundaries of a system, sources and destinations of data, flows of data, transformation processes, and stores of data for later use.

2.9 Summary and Conclusions

In this chapter some of the process representations, modelling methodologies and languages were reviewed and analysed to study the level of their capabilities to represent process specification requirements **table 2.3** shows the some of the process requirements that these process modelling methodologies can represent completely. Additionally they also represent other requirements partially. In this research study, how well a process representation or modelling methodology is capable of representing process characteristics is important since the aim of this study is to identify and represent construction process information using a formalised modelling methodology, for the purpose of process information exchange between software applications which deal with the manipulation of construction process information. The purpose is to identify a modelling methodology that can represent the complete set of construction process characteristics and hence support in the identification of the construction process specification requirements that need to be mapped to the PSL ontology in the investigation of the applicability of PSL in construction, and implementation of the language in a construction scenario

Formal expressions of an activity have proven to be good basis for developing systems (Benveniste and Berry 1991). A theory based on process algebra can even be developed for concurrent and communicating process specifications and proving some of their properties (Hoare, 1985; Milner, 1980) In principle, all kinds of constraints on activities (e g. triggering conditions or resources requirements) can be expressed as predicates However they remain so abstract and theoretical and cannot be used as communication languages among users For instance they do not differentiate between the control, information, and material flows as required by modelling. Furthermore they do not explicitly say anything about the termination or ending status of the activity (it would be necessary to analyze the output vector to determine it) Diagramming languages are better representations for functional aspects of processes.

Several graphical representations have been proposed to represent activities and processes (Vernada F B., 1996). For example the conventional box representation of SADT or EDEF0 ICOM box are among the many others. The graphical representation is preferred and has become the most wildly used method because it represents a good compromise between all other representations described above.

Form this study it is revealed that different techniques represent different views and aspects of process information and some representations are general approaches for the representation of a general process information while others are more specialized with more detailed process characteristics representation capabilities. The ability of modeling methodologies to capture process requirements is limited by the way the process and information is represented.

All the general systems, functional, and behavioral modeling methodologies reviewed in this chapter focus on the syntax of process specification rather than the semantics, or meanings, of terms. This may be sufficient when process information exchange is occurring within a single domain. However, exchange of process models data among different domains creates situations where the same terms can have different meanings. A process representation for information exchange between different applications or domains must have an unambiguous and a formal specification of the semantics of its terminologies. This capability can be achieved through the knowledge representation techniques or ontology modeling methodologies such as the IDEF5 and KIF techniques. Ontology provides a common terminology that helps to capture key distinctions among concepts in different domains/applications, which aids in the translation process

The Overall results of the literature review has shown that among the process modeling methodologies, which focus on the syntax of processes, the IDEF3 Process Capture Method best meets most of the process characteristics representation requirements. IDEF3, like others is essentially a diagramming/ graphical language, but provides more graphical constructs and a set of forms (elaboration forms) for capturing any kind information about the processes and participant objects involved. These constructs increased the capability of the methodology in capturing the required characteristics of a process.

In this research study few of the available process modeling methodologies were revived and a brief discussion is presented However, a detailed discussion of results from the analysis of twenty-six process representations studied in the second phase of PSL project is available in literature (Knutilla, A., et. al , 1998). There is also a summary of a subset of this analyses that included the DARPA/Rome Laboratory Planning Initiative work (Polyak et al 1997)

Methodologies and Techniques	**Information representation Views**	**Represent completely the following Process Data**
DADT	General Systems Modelling (GSM)	Data flow between processes, input/output/control & mechanisms, activities in the cases of transformation, data in the case of information,
IDEF0	Functional Modelling	Cost Data, Resource, Product Characteristics, Resource, Resource Requirements for a Task, Simple Task Representation and Characteristics, Task Executor, Alternative Task, Complex Task, Representation and Parameters, Conditional Tasks, Pre- and Post processing Constraints, State Existence Constraints, Convey the Ancestry or Class of a Task, Information Exchange Between Tasks, Support for Simultaneously Maintained Associations of Multiple Levels of Abstraction,
IDEF3	Process Description Capture Method	Resource Requirements for a Task, Simple Sequences, Simple Task Representation and Characteristics, Task Executor, Simple Precedence, Composition / Decomposition, Incompleteness /Vagueness, Associated Illustrations and Drawings, Complex Groups of Tasks, Complex Sequences; Complex Task Representation and Parameters, Concurrent Tasks, Conditional Tasks, Constraints, Implicit/Explicit Resource Association, Iterative Loops, Parallel Tasks, Pre- and Post processing Constraints, Serial Tasks, State Existence Constraints, State Representations, Temporal Constraints, Complex Precedence, Eligible Resources, information Exchange Between Tasks, Support for Task/Process Templates, Support for Simultaneously Maintained Associations of Multiple Levels of Abstraction
PETI-NETS	Behavioural modelling	Simple Groupings, Simple Sequences, Simple Task Representation and Characteristics, Task Duration, Extensibility, Composition / Decomposition, Alternative Task, Complex Sequences; Iterative Loops, Parallel Tasks, Parameters and Variables, Pre- and Post-processing Constraints; Serial Tasks, State Existence Constraints, State Representations, Mathematical and Logical Operations, Synchronization of Multiple, Parallel Task Sequences,
IDEF5	Ontology Description Capture Method	Assertions, Classification, composition, relation, state, intuitive and natural form, meta concepts, kind characters & attributes, decomposition, processes, state, sub kind,
KIF	Knowledge representation	Extensibility, Conditional Tasks, Parameters and Variables, Mathematical and Logical Operations,
DFD	Data flow modelling	Data flow between processes,

Table 2 3 Summary of the process modelling methodologies and the process data they represent completely

Chapter 3 Information Standardisation for Construction

3.1 Introduction

With the wide spread use of computer applications in the AEC/FM environment, information have become increasingly available in electronic format of representation for use in construction processes As a result there has been an increasing need for an electronic form of information exchange between construction applications in the sector. There are number of approaches for direct data exchange between construction software applications; Figure 3.1 (Foreses T. 1994) illustrates some alternative mechanisms for exchanging information. These include transferring physical data files, accessing a central database through standard access methods, and procedural calls through an application-programming interface. Ideally, all the three mechanisms can be used together (Luiten 94) These alternatives hold great promise for improved communications, information exchange and integration between computer applications. However they may not support the exchange of the exact concepts of information due to the differences in the meaning of information representations between the applications i.e. the information at source may differ in meaning at destination. It can be said that information exchange is consistent if the semantics of the information can be shared but not the message or syntax of the information alone. The ability to share accurate information between computer applications is an important capability for computer-based collaboration in construction. A primary requirement for such collaboration is the development of information representation standards that are needed to provide a unifying neutral language for representing the exchanged information. Standards allow the interpretation of information that can be shared through one or all of the three alternative mechanisms. Hence there is a need for standardization of information representation structures for the construction industry to support the exchange of meaningful construction information between diverse software applications. That is there is a need for a mechanism that can interpret the meaning of set of messages exchanged.

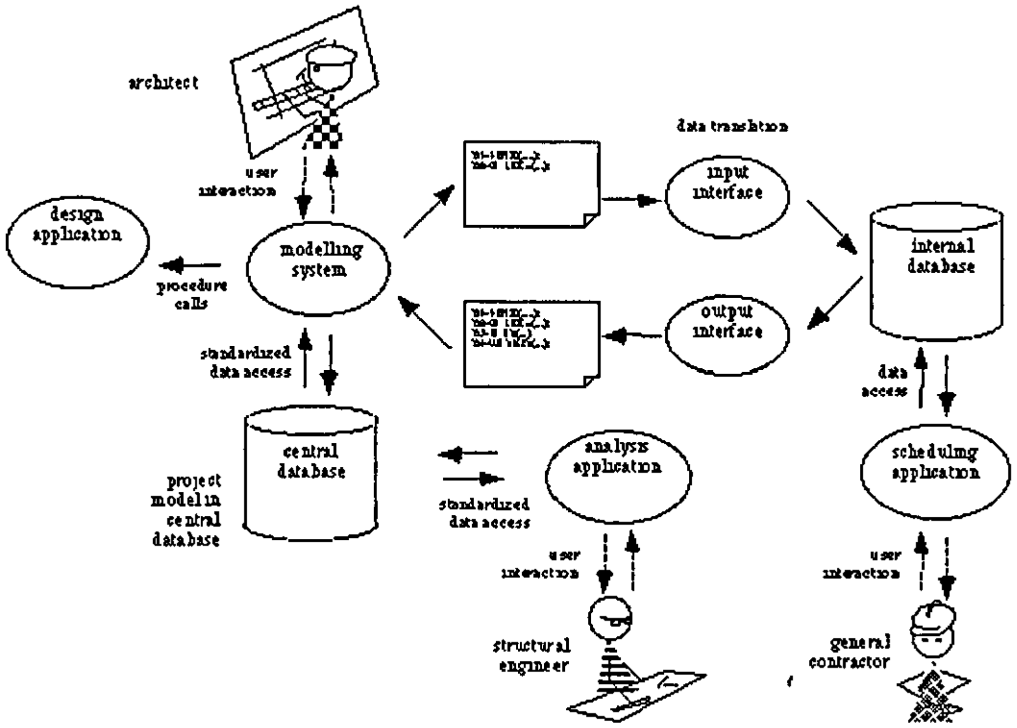


Figure 3 1 Alternative mechanisms for direct electronic information exchange (Luiten 94)

There have been several efforts in information standardisation in the field of industrial automation and integration in general. Most of the standardisation efforts in the AEC/FM sector have focused on facilitating product data and product related information exchange between software applications based on product data technology. In this chapter following a brief description of the concepts and objectives of information standardisation, literature from some of the major standardisation efforts for AEC/FM organisations and some of the available standards for construction industry are reviewed followed by conclusions.

3.2 Information Standardisation Definitions and Objectives

The problem of information sharing between software applications can take different approaches (Uschold, 1996):

- **Use of sharing services via point-to-point translation:** In which two or more systems share knowledge via run-time interactions. This approach is based on a "community of experts" metaphor, in which case one system will call on another to solve a problem, rather than request the knowledge to solve it itself.

- **Neutral interchange formats:** In which knowledge, and more generally, information, is exchanged between systems via an intermediate, "neutral" format. The exchanged information may be both items of static data, or "rules" of some kind whose primary interpretation has dynamic or behavioral properties.
- **Neutral authoring:** In which a neutral intermediate language is used for authoring, rather than exchanging, knowledge.

In all of these approaches, the issue of translation between the various underlying ontologies and representations plays a major role. These three approaches differ significantly in their cost (both immediate and long-term), scale, usability, and maintainability. At present, the most common approach to information sharing is through the use of point-to-point translators, converting between different formats. The "point-to-point", refers to a *direct* translation from a source format to a target format. An alternative approach, that is becoming increasingly common, is translation from source to target through a neutral interchange format. This is sometimes referred to as a "hub-and-spoke" model. This approach consists of translating each format first into the neutral format, and then from there, out to the target format. The three models of information and knowledge sharing are illustrated in **Figure 3.2.** The shaded boxes denote translators between the various representation languages.

3.2.1 Neutral Interchange Format

This approach requires:

1. Design of a sufficiently expressive neutral interchange format
2. Construction of two-way translators between the neutral format and each target application format

This model of information sharing aims to allow all applications to use information from all other applications, with several potential benefits. First, there is no need for an application developer to learn a new language for authoring, since the authoring takes place in the original application formats. Second, the different systems can be maintained independently: at least in theory, the only thing requiring changing should the application language be modified would be the translators to/from one's own format to the neutral format. Finally, there are potential savings to be gained by building fewer translators.

One needs to build O(n) translators instead of O(n^2) which would be required if point-to-point translators were built for every pair of applications. The potential benefit of requiring fewer translators may be more than offset by the expense of building the neutral format, especially where N, number of applications is not very large. Experience shows that this is very time-consuming, and therefore costly. If this cost is born by public funding, or is shared among major industrial or academic consortia, then there is more hope for the cost being absorbed in time among many users.

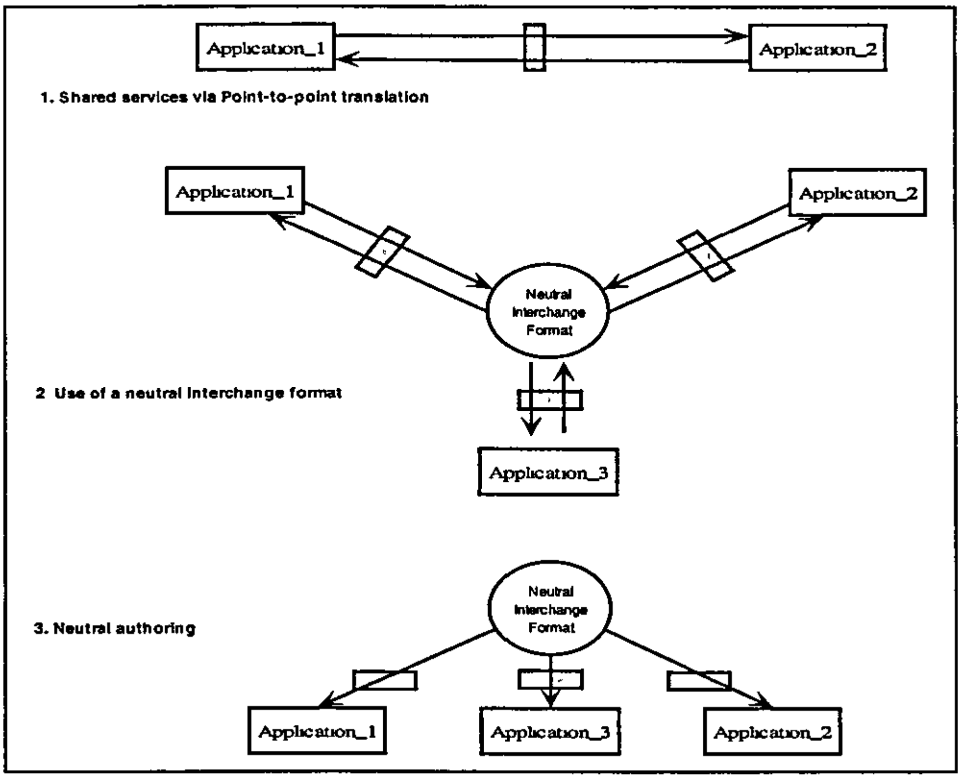

Application_1
Application_2
1. Shared services via Point-to-point translation
Application_1
Application_2
Neutral Interchange Format
2 Use of a neutral Interchange format
Application_3
Neutral Interchange Format
3. Neutral authoring
Application_1
Application_3
Application_2

Figure 3.2 Three different models of knowledge sharing (Uschold, 1996)

The other potential benefit of the neutral interchange format approach is the simplicity of the maintenance problem, when new formats come on line, or if existing ones change. In principle, one need only be concerned with translators to and from the interchange format and one's own application. If a new application comes on line, then once the translators are in place for the new application, then, ideally, no more work needs to be done by any of the other application developers/maintainers. This may be true as long as the interchange format remains stable, but substantial changes in target formats, may require the interchange format itself to be updated, which undermines this benefit

The increased risk of information lost in two translations compared with a single translation performed by a purpose-built point-to-point translator. This approach is used by several standards in the domain of Information and Communication Technologies, among which the STEP-family of standards (ISO 10303, 1994), presented in this chapter.

3.3 Information Representation Standardisation for AEC/FM: State-of-the-Art

There are numerous standardisation initiatives on various aspects of standards/information, and research projects aligned with some of the main efforts such as the STEP and IFCs, for information exchange in/for the construction industry. In the following section, a detailed literature of the some of the main international efforts (the STEP and IFCs) is presented, followed by a brief review on some of the available types of standards for construction industry, that can be found in the internet.

3.3.1 The ISO 10303 STEP Standard: Product Data Oriented Representation

The ISO TC184 is one of the one hundred and eighty eight committees of the ISO (International Standardization Organizations, Geneva, CH). Its scope is: "Standardization in the field of industrial automation and integration concerning discrete part manufacturing and encompassing the applications of multiple technologies, i.e. information systems, machines and equipments and telecommunications" (http://www.iso.ch). The standards developed within this scope are applicable to manufacturing and process industries, to all sizes of business, for extending exchanges across the globe through e-business. The standards developed within the ISO TC184 cover various domains related to industrial automation and integration, amongst which: enterprise modeling, enterprise architecture, communications and processes, integration of industrial data for exchange, access and sharing, life cycle data for process plants, manufacturing management, mechanical interfaces and programming methods, part libraries, physical device control, process specification language, product data, and robots for manufacturing environment. The ISO 10303 STEP standard was developed by the ISO Technical Committee on Industrial Data and Processes (TC184) (http://www.iso.ch/meme/TC184.html), Subcommittee on Industrial Data (SC4) (http://elib.cme.nist.gov/sc4/secretry.htm). In addition to STEP, SC4 is responsible for two other related standards: ISO 13584, Standard for Part Libraries (http://elib.cme.nist.gov/pub/sc4/www/plib.htm) and MANDATE, a standard for manufacturing management data (http://elib.cme.nist.gov/pub/sc4/www/mandate htm). Work within SC4 on STEP is carried out by a number of working groups. Among these is Working Group 3 (WG3), which is responsible for the development of the actual product models, and within WG3 is Team 12, which is responsible for Architecture, Engineering, and Construction (AEC) activities. This encompasses the areas of Offshore (Group 1), Shipbuilding (Group 2), Process Plant (Group 3), and Building Construction (Group 4). **Figure 3.3** shows the organisation of STEP. Work on STEP also overlaps with many other organizations for example the US *National Institute of Standards & Technology* (NIST) (http://elib.cme nist.gov/) supports many aspects of STEP development such as the National PDES Test bed. The *Product Data Exchange using STEP (PDES)* is the name of the STEP project within the US PDES work is carried out through the IGES/PDES Organization (IPO) http://elib cme.nist gov/pub/nipde/orgs/ipo.html under the parent organization of the US Product Data Association (US PRO) http://elib.cme nist.gov/pub/nipde/

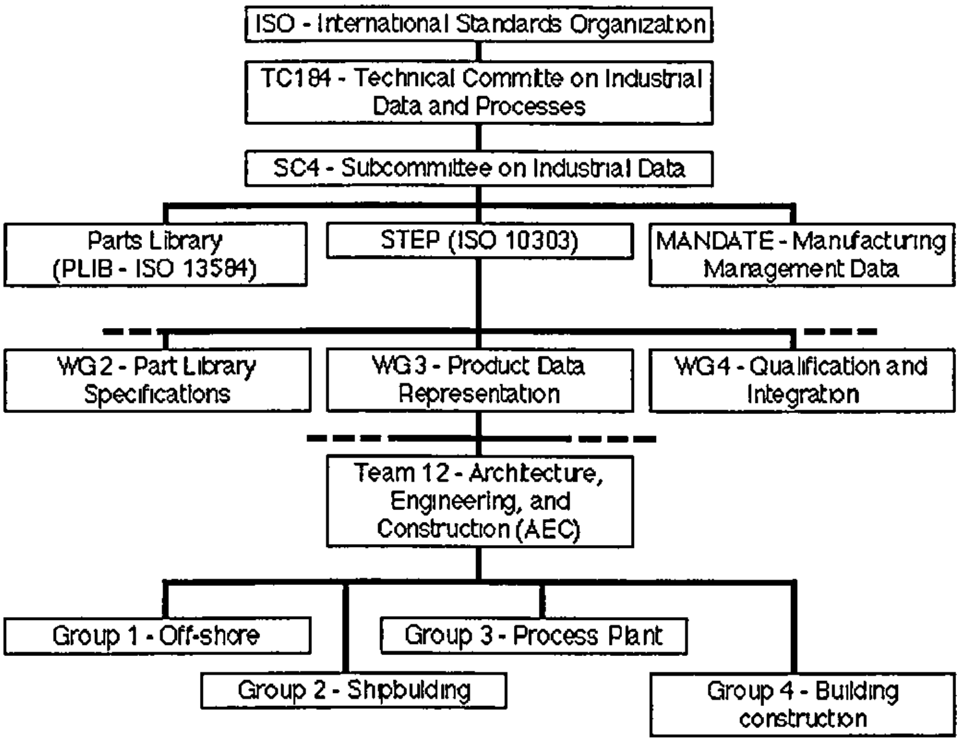

ISO - International Standards Organization
TC184 - Technical Committe on Industrial Data and Processes
SC4 - Subcommittee on Industrial Data
Parts Library (PLIB - ISO 13584)
STEP (ISO 10303)
MANDATE - Manufacturing Management Data
WG2 - Part Library Specifications
WG3 - Product Data Representation
WG4 - Qualification and Integration
Team 12 - Architecture, Engineering, and Construction (AEC)
Group 1 - Off-shore
Group 3 - Process Plant
Group 2 - Shipbulding
Group 4 - Building construction

Figure 3.3 the organisation of STEP.

3.3.1.1 The STEP Model Development Methodology

The development of standard model in STEP follows the production of standard product models for use within specific areas of application, called *Application Protocols (AP's),* and coordination of these models across application areas

The AP's grow out of specific industry needs, and the role of the AP is documented in an *Application Activity Model (AAM).* The AAM identifies the processes in which the AP is used, and shows the information flows among the processes using IDEF0 notation, which lists activities and the information flows between them as shown in **Figure 3.4.**

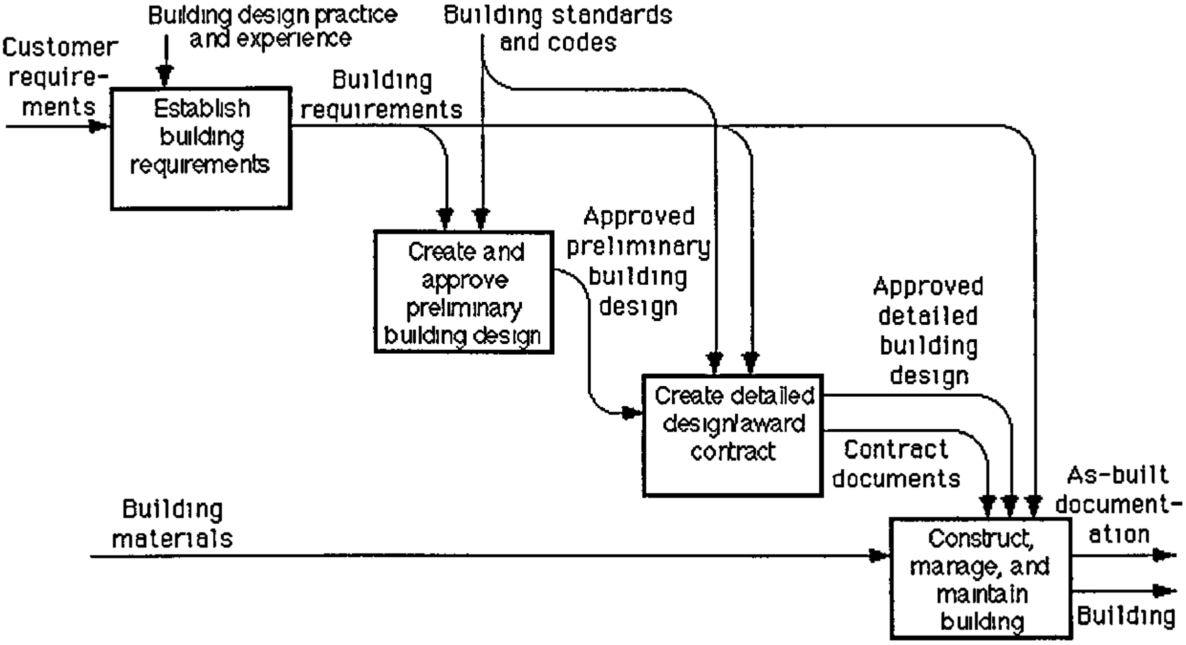

Building design practice and experience
Building standards and codes
Customer require-ments
Establish building requirements
Building requirements
Create and approve preliminary building design
Approved preliminary building design
Approved detailed building design
Create detailed design/award contract
Contract documents
As-built document-ation
Building materials
Construct, manage, and maintain building
Building

Figure 3 4 Example of an Application Activity Model (AAM) in IDEF0 Notation (from AP225 ISO 1995)

The AAM serves as an interface with industry participants in the modelling process and is the primary tool for determining the use of the model. An *Application Reference Model (ARM)* is developed that depicts the information that needs to be included in the AP using the terminology and concepts of the application domain. Finally in the integration process a model is developed that fully defines all the necessary data representation structures in a way that is compatible with other parts of the STEP standard. This is the *interpretation* process and it results in an *Application Interpreted*

Model (AIM) which draws upon the ARM and other reference models, either STEP-wide or *Integrated Generic Resources (IGR's)* and *Integrated Application Resources (IAR's).* In addition, *Conformance Classes* are defined and suites of test data are developed through which implementations can be tested. Finally, where the interpretation process leads to the same basic concepts being represented in two or more AIM's, these model segments are defined in an *Application Interpreted Construct (AIC)* for use in future AIM's. A fully developed AP specifically the AIM model presented in EXPRESS language is intended to be implemented to support information exchange.

3.3.1.2 Main features of the standard

Each part of ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data (Kemmerer, 1999). The objective is to provide a neutral mechanism capable of describing product data throughout the lifecycle of a product, independent from any particular system. The nature of this description makes STEP suitable not only for neutral file exchange, but also as a basis for implementing, sharing product databases, and archiving (ISO IS 10303-1). The ISO *Initial Graphics Exchange Specification (IGES)*(http://elib.cme.nist gov/pub/nipde/stds/wh-iges.html), a neutral data format for exchanging information among CAD systems has acted in many ways as a precursor for STEP. STEP was designed to be the successor of exchange standards as such the IGES, SET, and VDA-FS with notable difference that STEP was intended to support data sharing and data archiving which is more than supporting exchange of product data These distinguishing concepts are:

Product data exchange: the transfer of product data between applications. STEP defines the format of the product data to be transferred between applications. Each application holds its own record of the product data in its own preferred format. The data conforming to STEP is transitory and defined only for the purposes of exchange.

Product data sharing: the access of, and operation on, a single copy of the same product data by more than one application, potentially simultaneously. STEP is designed to support the interfaces between the single record of the product data and the applications that share it. The applications do not hold the data in their own preferred forms. The architectural elements of STEP may be used to support the realization of the shared product data itself. The product data of prime interest in this case is the integrated product data and not the portions that are used by the different applications.

Product data archiving: the storage of product data, usually long term. STEP is suitable to support the interface to the archive. As in product data sharing, the architectural elements of STEP may be used to support the development of the archived product data itself. Archiving requires that the data conforming to STEP for exchange purposes is kept for use at some other time. This subsequent use may be through either product data exchange or product data sharing

Early in the development of ISO 10303, SC4 recognized that the scope of the standard was extremely large. This fact resulted in a couple of fundamental assumptions that shaped the architecture of STEP. SC4 assumed it is unlikely that any organization would implement the entire ISO 10303, due to its wide scope. Therefore, the standard was divided into parts, in order for organizations to implement only the parts need to meet their requirements. Another primary concept contributing to the architecture is that the content of the standard is completely driven by industrial requirements. These and the concept of re-use of data specifications are the basis that led to the development of two distinct types of data specifications for standards. The first type, reusable, context independent specifications, are the building blocks of the standard. The second type, application-context-dependent specifications (application protocols) are developed to satisfy clearly defined industrial information requirements. The combination of these two specifications enables avoiding unnecessary duplication of data specifications between application protocols.

3.3.1.3 Components of ISO 10303 STEP Standard

The architecture of STEP is intended to support the development of standards for product data exchange and product data sharing. The industrial requirements and concepts of reuse have contributed to the evolution of the architecture over the past decade. The architectural components of STEP are reflected in the decomposition of the standard into several series of parts. The STEP document composition was developed at the June 1989 meeting of ISO TC184/SC4/WG1 as a series of parts. Each part series contains one or more types of ISO 10303 parts. **Figure 3.5** shows the structure of the STEP documentation. Each of the structural components and functional aspects of the STEP architecture are described as follows:

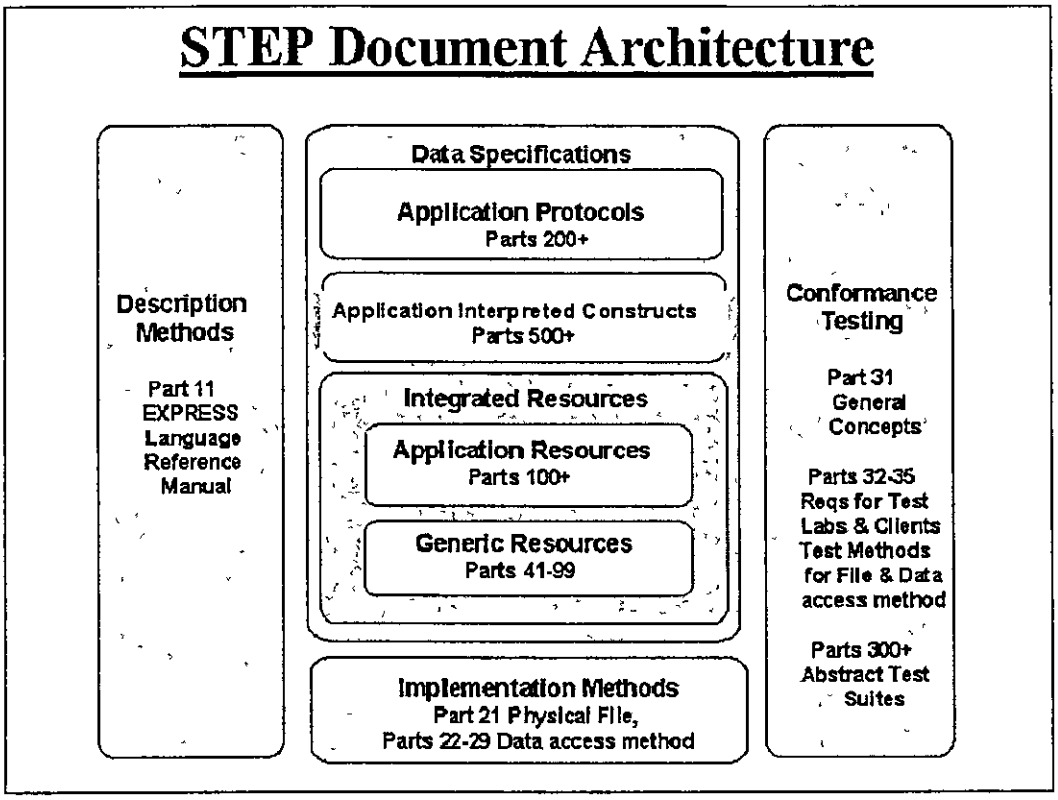

STEP Document Architecture
Description Methods
Part 11 EXPRESS Language Reference Manual
Data Specifications
Application Protocols
Parts 200+
Application Interpreted Constructs
Parts 500+
Integrated Resources
Application Resources
Parts 100+
Generic Resources
Parts 41-99
Implementation Methods
Part 21 Physical File,
Parts 22-29 Data access method
Conformance Testing
Part 31 General Concepts
Parts 32-35 Reqs for Test Labs & Clients Test Methods for File & Data access method
Parts 300+ Abstract Test Suites

Figure 3.5 Overview of the STEP document architecture (Kemmerer, 1999)

Description Methods: The first major component is the description method series of STEP parts. Description methods are common mechanisms for specifying the data constructs of STEP. Description methods include the formal data specification language developed for STEP, known as EXPRESS (ISO 10303-11:1994). EXPRESS is a Part of STEP and has been published with the number ISO 10303-11. Other description methods include a graphical form of EXPRESS, a form for instantiating EXPRESS models, and a mapping language for EXPRESS.

Implementation Methods: The second major component of STEP is the implementation method series of the 10303 parts. Implementation methods are standard implementation techniques for the information structures specified by STEP data specifications intended for implementation, application protocols (AP) Each STEP implementation method defines the way in which the data constructs specified using STEP description methods are mapped to that implementation method. This series includes the physical file exchange structure (ISO 10303-21:1994), the *Standard Data Access Interface (SDAI)* (ISO 10303-22:1998), and its language bindings (ISO 10303-23(DIS); ISO10303-24 (CD); ISO10303-26 (CD)) Implementation methods are standardized in the ISO 10303-20 series of parts.

Conformance Testing: The third major architectural component of STEP is in support of conformance testing. Conformance testing is covered by two series of the 10303 parts: conformance testing methodology and framework, and abstract test suites. The conformance testing methodology and framework series of the 10303 parts provide an explicit framework for conformance and other types of testing as an integral part of the standard. This methodology describes how testing of implementations of various STEP parts are accomplished.

Data Specifications: The final major component of the STEP architecture is the data specifications shown in **Figure 3.6.** There are four part series of data specifications in the STEP documentation structure, though conceptually there are three primary types of data specifications: integrated resources, application protocols, and application interpreted constructs All of the data specifications are documented using the description methods.

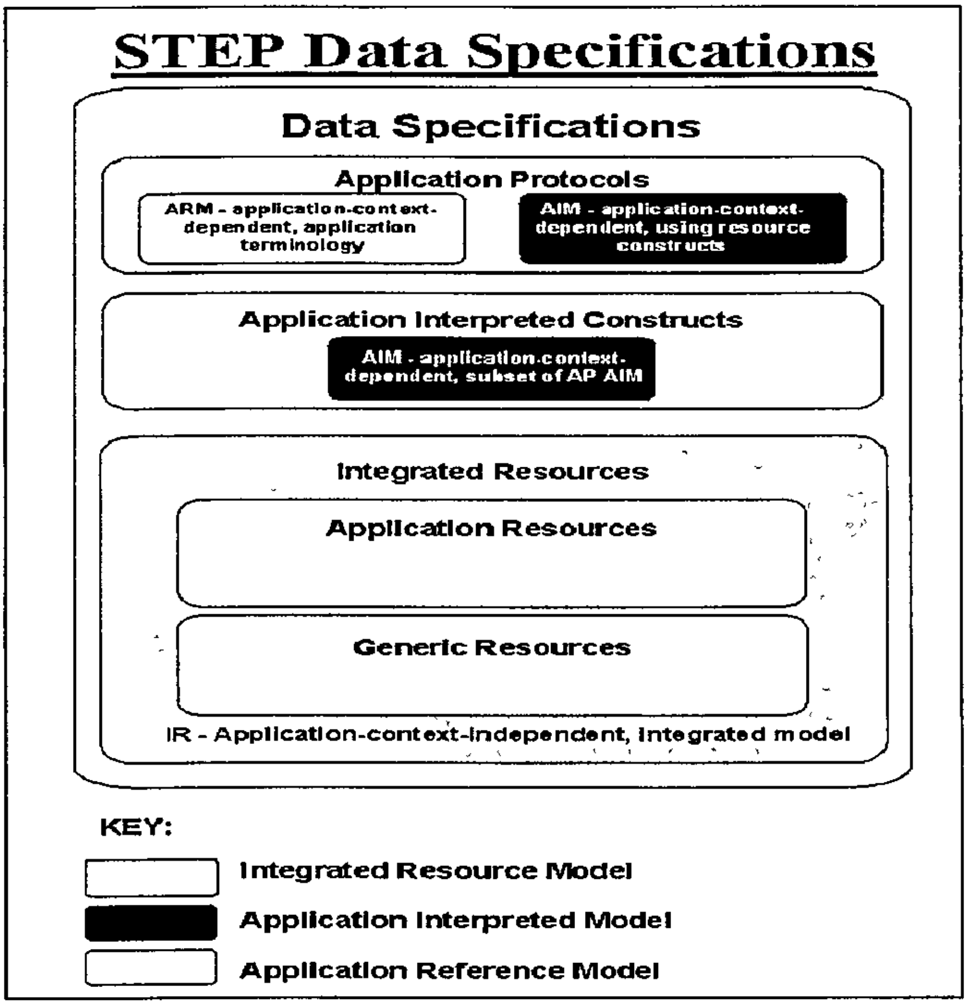

STEP Data Specifications
Data Specifications
Application Protocols
ARM - application-context-dependent, application terminology
AIM - application-context-dependent, using resource constructs
Application Interpreted Constructs
AIM - application-context-dependent, subset of AP AIM
Integrated Resources
Application Resources
Generic Resources
IR - Application-context-independent, integrated model
KEY:
Integrated Resource Model
Application Interpreted Model
Application Reference Model

Figure 3 6 STEP data specification (Kemmerer 1999)

Integrated Resources· The integrated resources constitute a single, conceptual model for product data. The constructs within the integrated resources are the basic semantic elements used for the description of any product at any stage of the product lifecycle. Although the integrated resources are used as the basis for developing application protocols, they are not intended for direct implementation. They define reusable components intended to be combined and refined to meet a specific need. The integrated resources comprise two series of parts, the integrated generic resources and the integrated application resources. The two series have similar function and form: they are the application, context-independent standard data specifications that support the consistent development of application protocols across many application contexts.

Application Protocols: Application protocols (APs) are the implementable data specifications of STEP. APs include an EXPRESS information model that satisfies the specific product data needs of a given application context. APs may be implemented using one or more of the implementation methods They are the central component of the STEP architecture, and the STEP architecture is designed primarily to support and facilitate developing APs. Many of the components of an application protocol are intended to document the application domain in application specific terminology. Application protocols are standardized in the ISO 10303-200 series of parts.

Application Interpreted Constructs: Application interpreted constructs (AICs) are data specifications that satisfy a specific product data need that arises in more than one application context. An application interpreted construct specifies the data structures and semantics that are used to exchange product data common to two or more application protocols Application protocols with similar information requirements are compared semantically to determine functional equivalence that, if present, leads to specifying that functional equivalence within a standardized AIC. This AIC would then be used by both application protocols and available for future APs to use as well. STEP has a requirement for interoperability between processors that share common information requirements. A necessary condition for satisfying this requirement is a common data specification. Application interpreted constructs provide this capability Application interpreted constructs are standardized in the ISO 10303-500 series of parts.

A new concept, "common resources" has appeared within the STEP community, which is aimed at maximizing the re-use of existing elements, either directly within the data specifications, or by means of the development of "application modules". Further information about the components of the standard; details of the components of the standard, and application modules (schema) of the ISO 10303 STEP standard on a page are available in (http //www nist gov/sc5/soap/). These depict, the organization and status of STEP Parts; the status of application modules; and explanatory text about the STEP respectively

3.3.2 The IAI's IFC de facto Standard: Product Data Oriented Representation

The International Alliance for Interoperability (IAI) is an international consortium of regional chapters registered and listed as non-for-profit organizations in North America, United Kingdom, Germany, France, Scandinavia, Japan, Singapore, Korea and Australia. Currently the IAI has about 650 membership organizations world-wide, being construction companies, engineering firms, building owners and operators, software companies, and academic institutions. The vision of the IAI is to provide a universal basis for process improvement and information sharing in the construction and facilities management industries (IAI transp, 2001). The vision is supported by the IAI's mission statement that is to define, promote and publish the Industry Foundation Classes (IFC), which is a specification for sharing data throughout the project life cycle, globally, across disciplines and across technical applications. More information about the IAI is available at http //www.iai-international.org.

The IFCs are data sharing specification, written in EXPRESS (ISO 10303-11:1994), a dedicated formal language developed by ISO TC184/SC4 Contents according to IFC are currently exchanged between IFC compliant software applications using the Clear text encoding of the exchange structure, the STEP physical file (ISO 10303-21:1994). The scope of the IFC specification is the project life-cycle of construction facilities, including all phases as identified by generic process protocols for the construction and facilities management industries, such as demonstrating the need, conception of need, outline feasibility, substantive feasibility study and outline financial authority, outline conceptual design, full conceptual design, co-ordinated design, procurement and full financial authority, production information, construction, operation and maintenance. Development of IFC is guided by versions and releases, which extend the scope successively. The processes supported by the current IFC2x specifications are outline conceptual design, full conceptual design, co-ordinated design, procurement and full financial authority, production information, construction, operation and maintenance.

The target applications to exchange and share information according to IFC2x are: CAD Systems, HVAC design systems, Electrical design systems, Framework design and scheduling systems, Structural analysis systems, Energy simulation systems, Quantity take-off systems, Cost estimation systems, Production scheduling systems, Clash-detection systems, Product information providers, Steel and Timber frame construction systems, Prefabrication systems, stand-alone visualisation tools and others. Industry Foundations Classes IFC2x have been endorsed by the ISO organisation as the ISO/PAS 16793 in November 2002.

3.3.2.1 The IFC Model Architecture

The IFC Model Architecture consists of the following layers and the organisation and relationships between the layers is shown in **figure 3.10**:

- Resource Layer
- Core Layer
    - o Kernel
    - o Extensions
- Interoperability Layer
- Domain Layer

Resource Layer: Resources can be characterized as general purpose or low level concepts or objects that are independent of application or domain need but which rely on other classes in the model for their existence. For instance, geometry is a widely used resource whose specification is independent of the domain However, an object within a domain must be defined before its geometry can exist. Exceptions to this characterization include classes from the Utility and Measurement Resources that are used by other, higher-level resource classes.

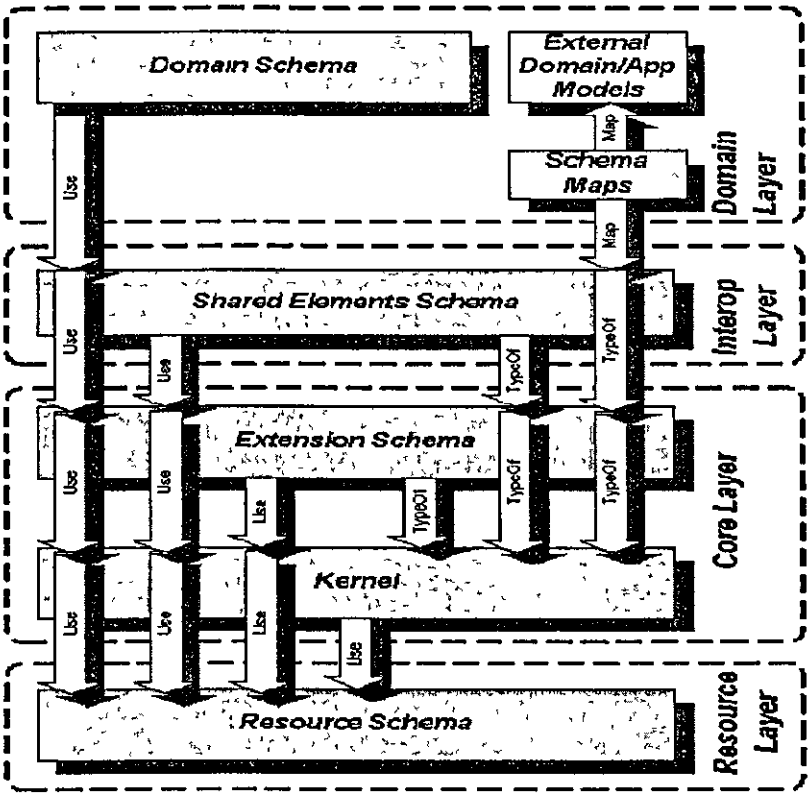


Figure 3.7 layering concepts of IFC architecture (IFC TG, 2000)

Core Layer: The Core forms the next layer in IFC Model Architecture. The Core layer provides the basic structure of the IFC object model and defines most general concepts that will be specialized by higher layers of the IFC object model. Core extensions from kernel classes are shown in **Figure 3.11.**

The Core includes two levels of generalization:

The Kernel: provides all the basic concepts required for IFC models within the scope of the current IFC Release. It also determines the model structure and decomposition. Concepts defined within the kernel are, necessarily, generalized to a high level. It also includes fundamental concepts concerning the provision of objects, relationships, type definitions, attributes and roles. The Kernel can be seen as a template model that defines the form in which all other schema within the model are developed (including all extension models). Its constructs are very general and are not AEC/FM specific, although they will only be used for AEC/FM purposes due to the specialization in the Core Extensions. The Kernel constructs are a mandatory part of all IFC implementations. The Kernel is the foundation of the Core Model. Kernel classes may reference classes in the Resource layer but may not reference those in the other parts of the Core or in higher-level model layers.

Core Extensions· provide extension or specialization of concepts defined in the Kernel. They are the first refinement layer for abstract Kernel constructs. More specifically, they extend those constructs for use within the AEC/FM industry. Each Core Extension is a specialization of classes defined in the Kernel and develops further specialization of classes rooted in the IfcKernel. Additionally, primary relationships and roles are also defined within the Core Extensions. A class defined within a Core Extension may be used or referenced by classes defined in the Interoperability or Domain layers, but not by a class within the Kernel or in the Resource layer. References between Core Extensions have to be defined very carefully in a way that allows the selection of a singular Core Extension without destroying data integrity by invalid external references. The goals for Core layer design are, definition of those concepts that are common to all parts of the model and that later can be refined and used by various interoperability and domain models; pre-harmonization of domain models by providing the set of common concepts; and stable definition of the object model foundation to support upgrade compatible IFC Releases

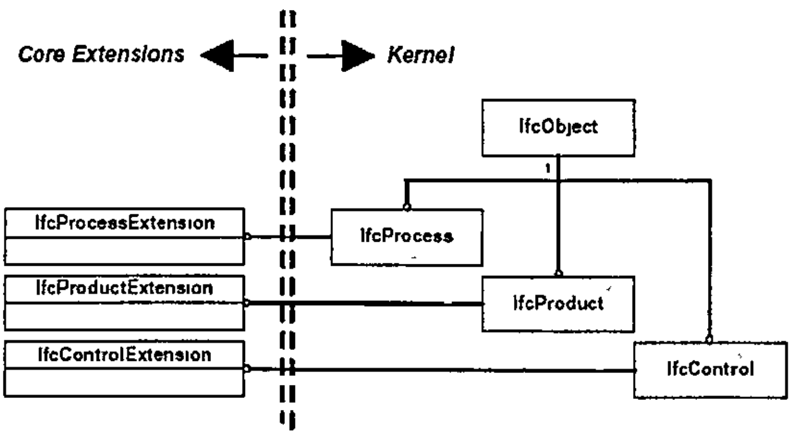


Figure 3.8 Core extensions from kernel classes (IFC TG, 2000)

Interoperability Layer: The main goal in the design of Interoperability Layer is the provision of schemata that define concepts (or classes) common to two or more domain models. These schemata enable interoperability between different domain models It is at this layer that the idea of a 'plug-in' model approach emerges. It is through the schemata defined at the Interoperability Layer that multiple domain models can be plugged into' the common IFC Core. The 'plug-In' approach also supports outsourcing of the development of domain models.

Domain Layer: Domain Models provide further model detail within the scope requirements for an AEC/FM domain process or a type of application. Each is a separate model that may use or reference any class defined in the Core and Independent Resource layers. Examples of Domain Models are Architecture, HVAC, FM, and Structural Engineering etc. An important purpose of Domain Models is to provide the 'leaf node' classes that enable information from external property sets to be attached appropriately.

3.3.2.2 The IFCs Architectural Principles

The IFC Model Architecture has been developed using a set of principles governing its organization and structure. These principles focus on basic requirements and can be summarized as (IFC TG, 2000):

- Provide a modular structure to the model.
- Provide a framework for sharing information between different disciplines within the AEC/FM industry
- Ease the continued maintenance and development of the model.

- Enable information modellers to reuse model components
- Enable software authors to reuse software components
- Facilitate the provision of better upward compatibility between model releases

3.3.2.3 The IFCs Architectural Model Schemata

The IFC Model architecture provides a modular structure for the development of model components, the 'model schemata'. There are four conceptual layers within the architecture, which use a strict referencing principle. Within each conceptual layer a set of model schemata are defined

1. The first conceptual layer provides Resource classes used by classes in the higher levels
2. The second conceptual layer provides a Core project model. This Core contains the Kernel and several Core Extensions.
3. The third conceptual layer provides a set of modules defining concepts or objects commonacross multiple application types or AEC industry domains. This is the Interoperability layer.
4. Finally, the fourth and highest layer in the IFC Model is the Domain layer It provides a set of modules tailored for specific AEC industry domain or application type

The architecture operates on a 'gravity principle'. At any layer, a class may reference a class at the same or lower layer but may not reference a class from a higher layer. References within the same layer must be designed very carefully in order to maintain modularity in the model design. Inter-domain references at the Domain Models layer must be resolved through 'common concepts' defined in the Interoperability layer. If possible, references between modules at the Resource layer should be avoided in order to support the goal that each resource module is self-contained However, there are some low level, general-purpose resources, such as measurement and identification that are referenced by many other resources.

3.3.2.4 Connecting External Models to the IFC Model

Fully harmonized IFC Domain Models are directly connected to the Core definitions. Domain Models that are not fully harmonized have to provide appropriate connection to relevant IFC class definitions in order to use the IFC model framework. Such models may be developed according to different technical architectures and methodologies but might need to be used in conjunction with the IFC model at some point. This can be achieved through the use of a connection mechanism The main requirements for connection are the facilitation of:

- Connection of externally developed, non-harmonized, Domain Models via a connection that provides a mapping mechanism down to Core and Interoperability definitions. The definition of the connection is in the responsibility of the Domain Model developer and is part of the Domain Model Layer.
- Establishment of an inter-domain exchange mechanism above the Core to enable interoperability across domains. This includes a container mechanism to package information. Therefore a connection is used where the definition of the connection is the responsibility of all Domain Models that share its use.

Connections are based on Core Extension definitions and enhance those Core Extension definitions. Those enhancements provide common concepts for all Domain Models that might further refine these concepts. As an example, the Building Element provides the definition of a common wall, whereas the Architectural Domain Model will enhance this common wall with its private subtypes and type definitions. A connection that is used by several Domain Models therefore provides a level of interoperability through shared connection definitions.

Non-IFC harmonized models can be connected to the IFC Core Model through a specifically defined mapping. For specific high-level inter-domain exchange that cannot be satisfied by common definitions in the Core, connection through mapping may provide a specific inter-domain exchange capability.

3.3.3 Classification Standard: (ISO TC59 SC13 TR 14177: ISO/PAS 12006)

Classification is a mechanism for categorisation of information according to specific criteria. The need for general classification systems grows with the increased internationalisation of the construction market and the rapid development towards a computer integrated construction process based on computer aided product data modelling (Ekholm, A. 1996). These processes require standardised ways of describing construction artefacts, and classification is a means to achieve this. Classification within the construction sector is based on pragmatic tradition and national needs, but internationally applicable classification tables must be founded on a neutral conceptual framework (Ekholm, 1996). The ISO Technical Report 14177 "Classification of information in the construction industry" aims at providing such a framework. The ISO Technical Report 14177 "Classification of information in the construction industry" (ISO 1994 a) is results of work Within ISO/TC59/SC13, a working committee of the International Standardisation Organisation, with the objective to develop principles for the building sector's classification system. The basic concepts in construction works classification within the ISO Technical Report that are of interest in the design, construction and management processes are as follows:

Facility or construction works: A building is defined as: "a type of facility comprising partially or totally enclosed spaces and providing shelter". An analysis of the examples of classifications of facilities in the report shows that they are based on four different kinds of properties The first three are functional, a) function with users, b) function with an installation, and c) function with an environmental agent, and the fourth, d), is based on intrinsic properties Intrinsic properties of the last category are load-bearing, enclosing, servicing, and spatial properties of the facility. These properties are used to support different kinds of functions.

A construction work: it is an artificial system, built for a purpose, it has a static ground construction, and relations to the environment like the surrounding nature and users. A construction work, as a whole, is a system of interacting parts, which may be divided into three main functional groups, load-bearing, enclosure (against for example climate and intruders), and servicing. Construction work parts interact and constitute systems of different kinds with new functions.

Space: a space is an object with certain geometrical enclosing properties and that it can have a function Spaces are defined as "Three dimensional spaces within and around buildings and other facilities, bounded actually or theoretically". The attributes in classification tables for spaces, represent functions in relation either to the users e g. "lavatory" and "dining-room", or a kind of installation e g. "boiler-room", or an external agent acting on the facility e.g. "rain"-shed. A *spatial relation* is a non-bonding separation relation among objects, and *space* is a set of spatially related objects. Spaces in a building are made up of building parts. These parts have spatial relations that constitute a suitable environment for user activities and objects Characteristic for spaces in buildings are their enclosing properties.

Element: it is "A physical part or system of a facility with a characteristic function (e g. enclosing, furnishing or servicing building spaces), defined without regard to the type of technical solution or the method or form of construction". The elements of a system are identified through a "top-down", functional, view. Three major kinds of elements according to the report are structure/enclosure elements; services engineering elements; and fixtures/equipment elements.

Designed element: is an element for which a "technical solution and form of construction" have been defined. For example, an "enclosure" element may be designed as a construction of gypsum board and studs, and then it is defined as a designed element. The concept of designed element is of importance for cost information and product modelling since it includes both functional and material properties.

Work section: it is "One or several physical parts of a facility, viewed as the result of particular skills and techniques applied to particular construction products and/or designed elements during the production phase". According to the definition, the concept work section has reference to the construction work part and its assembly. A work section is a construction result characterised by the used construction products, and their material substance, and the production activity. a work section is a "bottom-up" or compositional view of a physical part of a construction work.

Production activity: uses resources and produces results. The resources are construction products, construction aids and human effort (labour and thought), the results are both physical parts of construction works and other things or processes necessary during production Production activities are particular skills and methods in work, which transforms and assembles particular construction products into so called "work sections", results. The aim of the production activity is to achieve work sections with "element" properties.

Construction product: they are defined as: "Products, components and 'kits of parts' incorporated or intended for incorporation into facilities, including furniture and equipment". Construction products are things with the purpose to be used as, or transformed into, parts in construction works. Construction work parts and construction products may have the same composition and internal structure, the main difference is that the former is produced on site while the latter is produced "off site" with the intention to be assembled on the site.

Construction aids: are defined as "Scaffolding, formwork, machines and tools (including required energy), consumable stores, construction products used for temporary structures and facilities, and other objects needed for the purposes of the construction process which are not incorporated into and do not furnish or equip the facility"

Attribute: a specific table for attributes can be of use for "internal arrangement of technical documents, structuring of product data bases, structuring of other classification tables according to primary attributes, and definition of requirements for projects and resources generally". The attributes represent factual or phenomenal and intrinsic or mutual properties that the construction work has either by itself or in relation to some other thing, for example a user or a reference frame. The types of attributes that are of interest to the construction industry are: performance, function, shape, location, material, price, and production time.

3.3.3.1 Conceptual Schema of Construction Works

Figure 3.12 shows a schema presented in the ISO Technical Report that relates basic concepts for describing construction works. The schema shows a level order with buildings in the highest level followed by the levels of elements, work sections and construction products in successively lower levels. These are all seen as produced physical objects with examples of different attributes listed. The schema is developed according to the NIAM information modelling technique (Nijssen G M. and Halpin T.A, 1989)

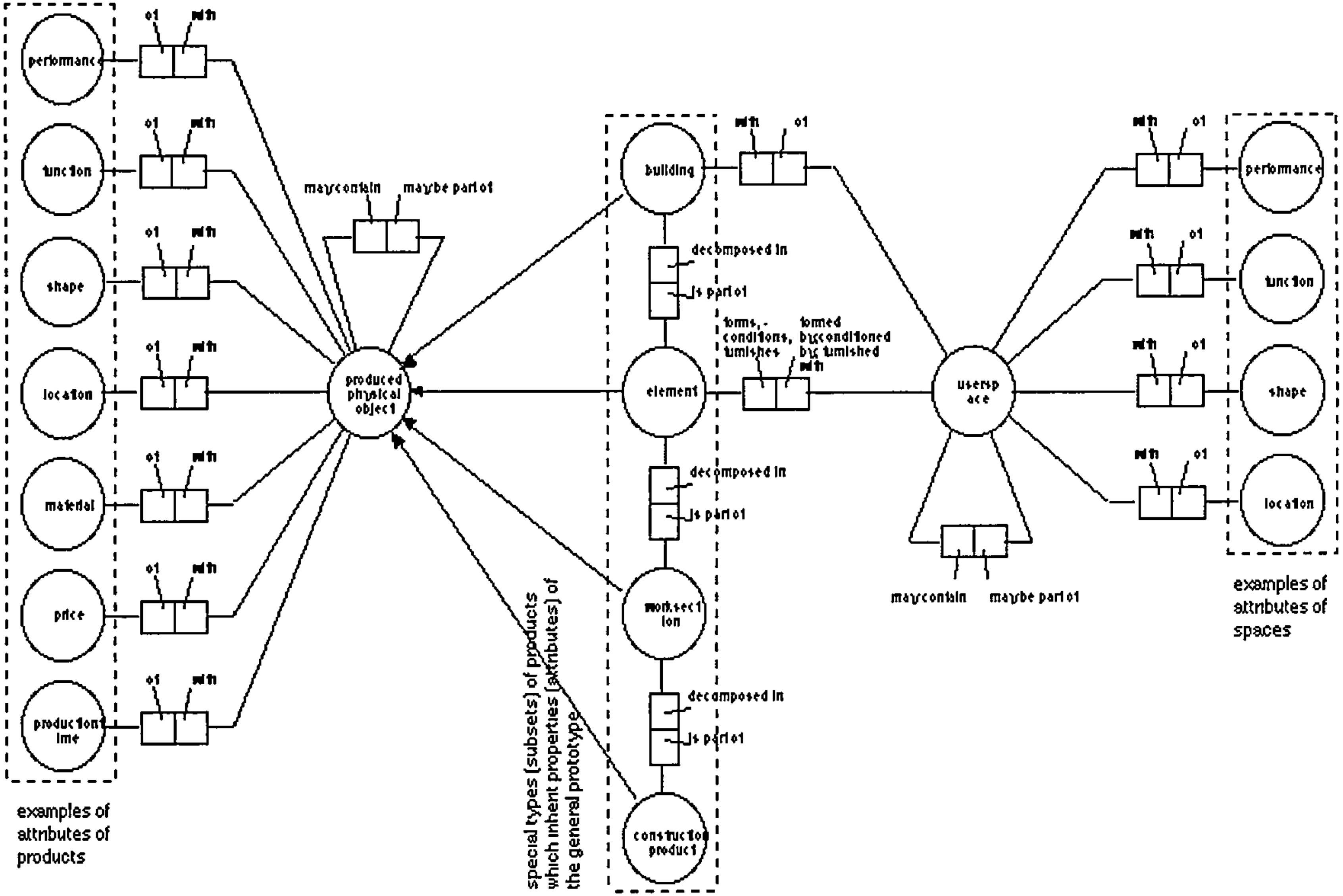


Figure 3 9 Schema in the ISO Technical Report relating basic construction information concepts (ISO Technical Report 14177)

Figure 3.13 shows a conceptual schema for construction works that relates some of the basic concepts discussed in the ISO Technical Report 14177. This schema is presented in EXPRESS-G, a graphical notation technique of the EXPRESS information modelling language. The construction artefacts are produced in the construction process are infrastructure units, construction works, construction work elements, element parts, and spaces. These have properties of specific interest to the construction process like production time, resource requirements, cost, etc. The functions of these are the relations to its environment, for example the site and the users The Classification of construction works Technical Report has been endorsed by the ISO organisation as the ISO/PAS 12006 in 2000.

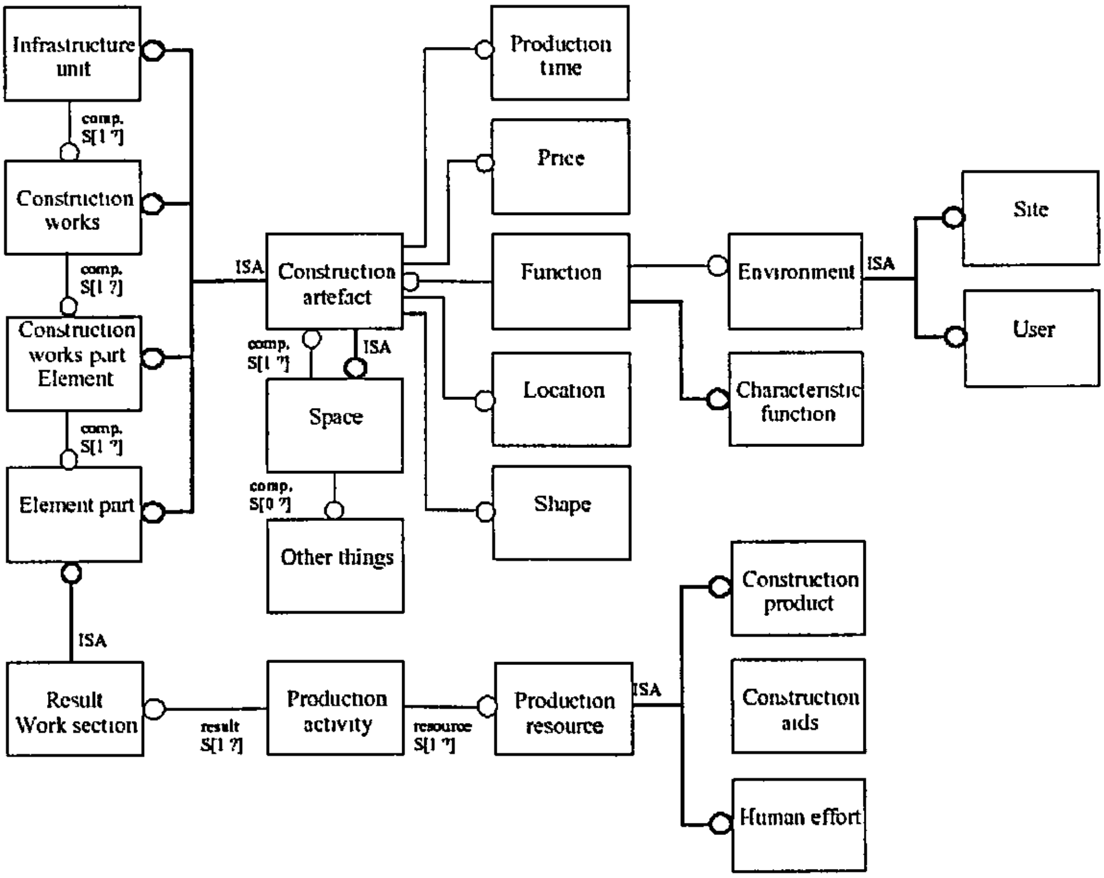


Figure 3 10 Level order and main properties of construction works (source:Ekholm, A. 1996).

3.3.4 Other Standardization Developments

Additionally, many other standardization efforts and developments have been underway (Froese. T , 1994):

- Several large pan-European research projects such as ATLAS, COMBINE, CIMSTEEL, COMBI, PISA, etc., are involved in developing integrated systems for the AEC industry, which may contribute to information standards.
- Several of the standardization efforts address information about the construction process in addition to the product (e g., construction activities in addition to the building components themselves). Examples include the Information Reference Model for AEC (IRMA), which is a generic model of construction processes that has been used as a vehicle for discussion among many international researchers. **Figure 3.14** showing some of the high-level data objects used to describe project information and the relations among them.
- Work is also being carried out in the standardization of information relating to document management for construction. The aim is to allow consistent access to and information about various construction documents, which may or may not be stored electronically.
- An ISO committee (TC10/SC8/WG13) is developing an international standard for the use of layers in computer-aided drafting (CAD) for construction. Their current proposal is that individual layers should be identified in terms of three mandatory information categories agent (the party responsible), construction element (type of component), and graphical element (colours, hatching, etc.) as well as four optional categories status (e g., existing structure to be demolished, new structure, etc.), time segment, space segment (location), and a user-defined category.
- Froese (1994) also has summarised various aspects of standards for information exchange in the construction industry that was the focus of discussion in "the standardization of information structures in the construction industry" workshop held at the Royal Institute of Technology, Stockholm, on April 6-7, 1994.

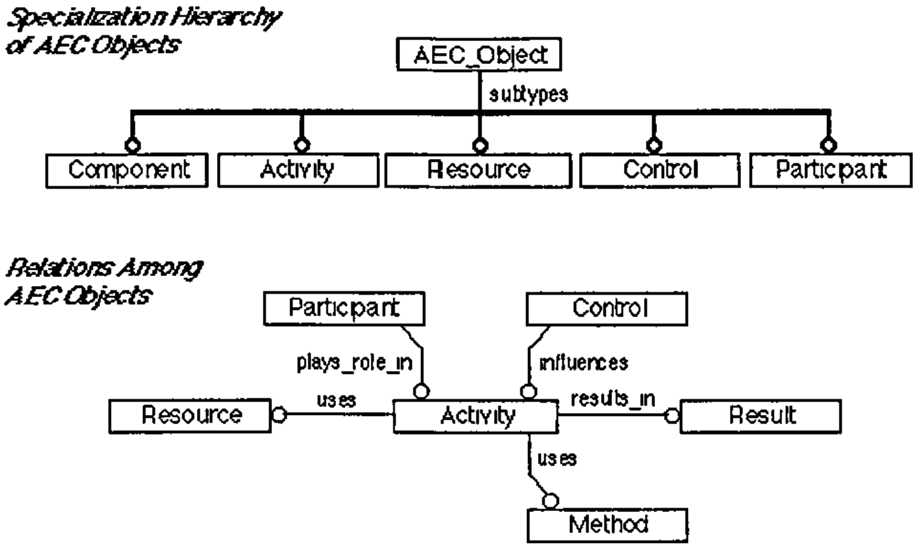


Figure 3 11 Portions of an information model for AEC projects (adapted from Froese 1994).

3.3.5 Some of the Available types of Standards in the Construction Industry

Numerous standards initiatives have been and many more are underway on different aspects of standards resulting in many different types of standards. In this section some of the available types of standards for construction industry that can be found in the Internet are presented including brief descriptions of their nature, use and origin. Additionally, A very comprehensive list of current and recent projects on standardization and information standards initiatives is available at:
http //www.cica org.uk/construction_IT_standards_links html.

CITE, Construction Industry Trading Electronically: CITE has developed a range of data exchange standards. These include enquiries, orders, invoices, and bills of quantities, valuations, and project information. CITE has also developed XML standards based upon existing international data standards, to allow companies across the construction industry to access the XML benefits. (http://www.cite.org.uk)

Industry Foundation Classes (IFC's): are classes of objects in an agreed manner that enables the development of a common language for construction for Interoperability among AEC/FM software applications across the domains. These classes provide a universal basis for process ımprovement and "product related ınformatıon" sharıng ın the construction and facılıtıes management industrıes The first Chapter of the International Allıance for Interoperabılıty was set up in North Amerıca ın June 1995, the UK Chapter ın January 1996. http://www.ıaı-ınternatıonal org

Data Interchange Format (DXF): The DXF file format was orıginally created by AutoDesk to represent 3D models and scenes buılt wıth AUTOCAD. The DXF format ıs synonymous wıth the DWG format.

Coordinated Project Information (CPI): thıs ıs for coordınatıon of contract documents. Standards (BPIC, 19987) UK approach for cross-referencıng between the documents for ıntegratıon of the suıte:

- Drawings to Specıficatıons
- Bılls to Specıfications
- Not spec to drawıngs
- Not spec to bılls

Building Construction eXtensible Mark-up Language (BcXML): these are developed Under a European communıty funded project called "E-Construct" XML schema taxonomy and dıctıonary which ıs focused on construction products, resources, work methods and regulatıons. Thıs new communıcation technology aıms to provıde the industry wıth a powerful but low cost communicatıon ınfrastructure that support e-buısness between clıents, archıtects and engıneers, supplıers (of components, systems and services), contractors and sub contractors. It wıll integrate wıth e-commerce and design / engıneering applicatıons and support vırtual constructıon enterprıses over the borders of the ındıvıdual European states. **Http://www.econstruct.org**

aecXML: thıs ıs XML based language for representıng ınformation ın the AEC industry and facilıtate ınformation exchange on the Internet. The ınformatıon may be resources such as projects, documents, materıals, parts, organizations, professıonals or activities such as proposals, desıgn, estımatıng, schedulıng and constructıon. Thıs ınıtıatıve ıs allıed to the IAI and the aim is to bring IT to buılding design, constructıon and operation and, to develop an envıronment ın whıch computer programs can share and exchange data automatıcally (wıthout translatıon and human intervention), regardless of the type of software or where the data may be resıdıng. **Http://aecxml.org**

ebXML: Global XML framework for business specification to lower the barrier of entry to electronic business in order to facilitate trade, particularly with respect to small and medium sized enterprises and developing nations The UN / CEFACT The United Nations body for Trade Facilitation and Electronic Business and OASIS (Organisation for the Advancement of Structured Information Standard) initiate a world-wide project to standardize XML business specifications. The ebXML initiative was established to develop a technical framework that will enable XML to be utilised in a consistent manner for the exchange of all electronic business data, in order to create a single global electronic market. **(Http://www.ebxml.org)**

Business Applications Software Developers' Association (BASDA): BASDA has developed standards and accreditations for handling business-to-business and business to Government e-Commerce, euro compliance and VAT handling. **(Http://194.164.46.55/)**.

CIMsteel Integration Standards (CIS): Standard object definitions for structural steelwork. These define how data is represented, shared, and managed between the myriad of software applications used in the structural steelwork industry. CIS/2 is an extended and enhanced second-generation release of the CIMsteel Integration Standards (CIS), CIS/2 has been developed to facilitate a more integrated method of working through the sharing and management of information within and between companies involved in the planning, design, analysis and construction of steel framed buildings and similar structures. As this new method of working is likely to evolve gradually, the standards may be implemented in various degrees of complexity, from basic file exchange through to advanced implementation in a Database Management System (DBMS). For detailed information on CIS Refer to CIS/2 by STEP (ISO 10303) and http://www.cis2.org

AP225 (STEP): Standard object definitions for the exchange of building component geometry (ISO 1995). AP225 is aimed at representing buildings as assemblies of elements e.g., beams, columns, windows, etc. along with the explicit (i.e., non-parametric) 3D geometry of each element and some additional information such as material properties, building element classification or element versions. The AP has been developed as a German nationally funded project headed by W. Haas, and it is the furthest along in the standards process of the building construction AP's, having reached

Committee Draft stage in January 1996. Experimental implementations have been completed that exchange complex building CAD models between heterogeneous CAD systems

STEP CDS 2D (STEP): 2D drawing exchange for construction object based exchange to manage information

IFCXML: IFC in XML format. The goal of this project (the ifcXML extraction and evaluation project) is the provision of the internationally agreed content and structure of the IFC2x specification (and any valid subset thereof) to the XML community. This is to

1. Enable the exchange of IFC data files alternatively as XML instance documents

2 Enable the reuse of IFC content and structure within XML based initiatives for data exchange and sharing in the construction and FM industries **(Http://www.iai.org.uk)**

EDIFACT: ebXML is developed from EDIFACT. Electronic data interchange (EDI) is the electronic transfer of business data from one independent computer system to another using agreed upon standards of data format. In order to implement EDI within an industry, there must be well-defined standards and protocols for each type of message that can be exchanged. Organizations have been established to create these standards, including the following:

- UN/EDIFACT: overall organization for establishing EDI, a joint sub-group of the UN/ECE/WP.4/GE.1 and ISO/TC154. EDIFACT boards exist for Western Europe, Eastern Europe, North America, Australia/New Zealand, Japan/Singapore, and Africa.
- JM7: world-wide construction group within UN/EDIFACT.
- EDIBUILD: a Pan-European EDI user's group for the construction industry that looks at usage issues such as business requirements, implementation issues, liaison with other sectors, and trial use
- MD5: the technical body within the Western European EDIFACT Board responsible for message development and maintenance for the construction industry.
- CIAG, Construction Industry Action Group, a users group for EDI within the construction industry in North America.
- PDIX, Process Data Exchange Institute.

3.4 Summary and Conclusions

This chapter presented a literature review on industrial information standardisation, including the efforts, developments and outcomes and information representation and modelling methodologies used in the process of standardisation Additionally some of the available types of standards in the construction environment are briefly reviewed and summarised in **table 1.3.**

Information standards provide a neutral representation or language through which accurate meaningful information can be exchanged between users of integrated computer applications. Numerous standards have been developed and several are underway for industrial in general including construction industry. These standards support different aspects of projects information. Hence these standards were developed based on different information representation or modelling methodologies. Most of the standardisation developments and efforts have focused on product aspect of projects information and the product data technology and product modelling approach have been key technologies in these developments and efforts. However the process aspects of projects information has not received enough attention to date. In some of these efforts such as the IAI's IFCs' target applications include software packages, which deal with management of process information such as cost estimating and scheduling and planning. In the IFC models the integration of information related to process is based on the connection between the processes and product data that is from viewpoint of data requirements for these processes in producing the products However the IFCs have included some project management related objects in the IFC Release 2.0.

In addition to physical information about buildings, the Industry Foundation Classes (IFCs) tend to represent project management information such as estimating and scheduling data. The IFCs have included some project management-related objects since their initial Release, in addition to the core concepts relating to the project management portions of the IFC, a significant number of new and revised objects have been included in the IFC Release 2.0. However these have received no implementation until the Project Management (PM) Domain Group of the IAI North American Chapter (IAI NA PM) held a workshop to conducted trials of the project management portions of the IFCs in February 1999 (T. Froese, et al., 1999.) to test how well the IFCs represent the cost estimating and scheduling information and evaluate the current IFC models as they relate to the project management tasks of estimating and scheduling (Clayton et al 1998). Form this exercise it was identified that changes of class names between versions (from IFC Release 1.5.1 to IFC Release 2.0 Beta) were inevitable resulting confusion and required redundant work during the implementation of the PM test case.

The information modelling methodologies provide the basis for the information standardisation process by documenting the aspects of information that are the target in the standards development. Hence standards based on product models may not facilitate the complete exchange of construction projects information and no literature to date has stated that these standards can support the exchange of the entire aspects of projects information As a result the need for process information standards that would complement the product model based initiatives for interoperability of software applications in the construction industry become important. Hence a need for appropriate process information modelling methodology that is capable of documenting all the necessary process aspects of projects that are necessary for specifying construction processes provide the basis in the process of standardising process information representations.

Existing modelling methodologies support the representation of different aspects of projects. This implies that there will be a need for the use of combination of the existing modelling methodologies to support a completed representation of the projects information otherwise the development of different standards for different aspects of projects information will continue. This is because the use of different standards for example product data and process information standards by organisations could be expensive compared to the traditional method of work. Therefore, there is a need for these standards to merge or find a way for one standard to incorporate all the information need to be exchanged between software applications of organisations of an industry.

The aim of this research study is to identify and study the process aspects of construction projects information, as a first step by targeting information within the pre-construction process stages, and based on existing standards and standardisation efforts to develop a mechanism that would compliment the product model data exchange approaches for construction software interoperability.

Name of standard	Scope	Target Applications	Domains	Applications	Description Methods	Standardisation
ISO 10303 STEP	Industrial automation and integration	Information systems, machines and equipments and telecommunications	Enterprise modelling, enterprise architecture, communications and processes, integration of industrial data for exchange, access and sharing, life cycle data for process plants, manufacturing management, mechanical interfaces and programming methods, part libraries, physical device control, process specification language, product data, and robots for manufacturing environment	Manufacturing and process industries	EXPRESS (ISO 10303-11 1994) IDEF0	Conceptual model
Industry Foundation Classes (IFC)	AEC/FM	CAD Systems, HVAC design systems, Electrical design systems, Framework design and scheduling systems, Structural analysis systems, Energy simulation systems, Quantity take-off systems, Cost estimation systems, Production scheduling systems, Clash-detection systems, Product information providers, Steel and Timber frame construction systems, Prefabrication systems, stand-alone visualisation	Project life cycle, globally, across disciplines and across technical applications	Construction and facilities management industries	EXPRESS (ISO 10303-11 1994) IDEF0	Conceptual model
ISO TC59, 1993 Classification of Information in the Construction Industry	Construction projects information			Construction industry	NIAM information modelling technique	Categorisation and classification
Data Interchange Format (DXF)		Cad applications				Data format
Building Construction eXtensible Mark-up Language (BcXML)	Design / engineering applications and virtual construction enterprises	e-buisness between clients, architects and engineers, suppliers (of components, systems and services), contractors and sub contractors		Construction industry		Construction products, resources, work methods and regulations
aecXML	Building design, construction and operation	proposals, design, estimating, scheduling and construction		AEC industry		
CIMsteel Integration Standards (CIS)	Planning, design, analysis and construction of steel framed buildings and similar structures	Software applications in the structural steelwork industry				Standard object definitions for structural steelwork

Table 3.1 Summary of some of the available and under development standards for construction

Chapter 4 Information Standardisation for Manufacturing

4.1 Process Specification Language (PSL)

As the use of information technology in the manufacturing operations matured (Schlenoff, et al. 1999), and the capability of software applications to interoperate became increasingly important, motivated by the growing need to interoperate process related manufacturing applications, a Process Specification Language (PSL) was developed, at National Institute of Standards and Technology (NIST), for process information exchange between multiple manufacturing process related applications such as process modelling, process planning, scheduling, simulation, workflow, project management, and business process re-engineering tools by translating between their native format and PSL.

The Process Specification Language (PSL) Project at NIST is organized under the Manufacturing Systems Integration Division (MSID). For many years MSID has been involved in the definition of a neutral representation of product data that was realized recently through the STEP standard (http://www.mel.nist.gov/psl/). With that effort well underway, the representation of manufacturing process became another candidate area of focus Like product data, process data is also used throughout the life cycle of a product, from early indications of manufacturing process flagged during design, through process planning, validation, production scheduling and control. In addition, the notion of process also underlies the entire manufacturing cycle, coordinating the workflow within engineering and shop floor manufacturing.

The PSL project at the National Institute of Standards and Technology (NIST) has been addressing the issue of software interoperability difficulties by creating a neutral, standard language for process specification to serve as an interlingua to integrate multiple process-related applications throughout the manufacturing life cycle. This interchange language is unique due to the formal semantic definitions (the ontology) that underlie the language. All concepts in PSL are formally defined, using the Knowledge Interchange Format (KIF) (Genesereth, M., Fikes, R , 1992), to eliminate the ambiguity usually encountered when exchanging information among disparate applications. Because of these explicit and unambiguous definitions, information exchange can be achieved without relying on hidden assumptions or subjective mappings. Following a brief look at the scope and goals of the language a literature review of the development and components of PSL is presented in the following sections.

4.2 Scope of PSL

The scope of PSL project (C. Schlenoff, A Knutilla, S. Ray, 1997) was limited to the realm of discrete processes related to manufacturing, including all processes in the design and manufacturing life cycle. Business processes and manufacturing engineering processes are included in this work both to ascertain common aspects for process specification and to acknowledge the current and future integration of business and engineering functions.

4.3 Goal of PSL

The goal of PSL is to create a process interchange language that is common to all manufacturing applications, generic enough to be decoupled from any given application, and robust enough to be able to represent the necessary process information for any given application. This representation would facilitate communication among the various applications because they would all have a common understanding of concepts to be shared. A good analogy is that PSL is for discrete process data as STEP (ISO 10303, the Standard for the Exchange of Product Model Data) is for product data exchange between multiple applications.

Although the ultimate goal of PSL is for information exchange there are a number of different opinions regarding other ways in which it could be used (Schlenoff. C., 1999), these include: The high level concepts of PSL could be used to model process in multiple domains and integrate multiple domains at highest level of process. In addition, the goal of this project is to create a "process specification language", not a "process characterization language". The definition of a process specification language is a language with which to specify a process or a flow of processes, including supporting parameters and settings. This may be done for prescriptive or descriptive purposes and is composed of ontology and one or more presentations. This is different from a" process characterization language", which referred to as a language describing the behaviours and capabilities of a process independent of any specific application. For example, the dynamic or kinematics properties of a process independent of a specific process can be included in this characterization language.

4.4 PSL Related Works

PSL is a neutral language for process specification to integrate multiple process related software applications throughout the manufacturing process life cycle (from initial process conception all the way through to process retirement). This project is related to, and in many cases has been working closely with, many other efforts. These include individual efforts (those involving only a single company or academic institution) such as A Language for Process Specification (ALPS) Project (Catron, B., Ray, S , 1991), the Toronto Virtual Enterprise (TOVE) Project (Fox, M., et al, 1996), the Enterprise Ontology Project (Uschold, M., et. al., 1996), and the Core Plan Representation (CPR) Project (Pease, A., 1998) In addition, the PSL project is in close collaboration with various projects (those that involve numerous companies or academic institutions) such as Shared Planning and Activity Representation (SPAR) Project (Tate, A ,), the WorkFlow Management Coalition (WfMC) (Belgium, 1994), and the Process Interchange Format (PIF) Project (Lee, J., et al, 1998).

ALPS, was a NIST research project whose goal was to identify information models to facilitate process specification and to transfer this information to process control. PSL project, which could be viewed as a spin-off of the ALPS, is a project with a goal to take a much deeper look into the issues of process specification and explore these issues in a much broader set of manufacturing domains.

The TOVE project provides a generic, reusable data model that provides a shared terminology for the enterprise that each agent can jointly understand and use. The goal of Enterprise Ontology project is to provide "a collection of terms and definitions relevant to business enterprises to enable coping with a fast changing environment through improved business planning, greater flexibility, more effective communication and integration". While both TOVE and the Enterprise Ontology focus on business processes, there are common semantic concepts in both these projects and the manufacturing process focused PSL.

The aim of CPR project is to develop a model that supports the representation needs of many different military planning systems The SPAR project is an ARPI (ARPA (Advanced Research Projects Agency)/Rome Laboratory Planning Initiative) funded project whose goal is similar to CPR. Both of these projects are similar to PSL in the sense that the aim is to create shared model of what constitutes a plan, process, or activity and their core models have similar roots. However, SPAR and CPR focus more on military plans and processes

While, PSL solely focused on developing a neutral representation for exchanging manufacturing process information, PIF (Lee, J., et al, 1996) is an interchange format based upon formally defined semantic concepts, like PSL. However, unlike PSL, PIF is focused on modeling business processes and offers a single, syntactical presentation, the BNF (Backus-Naur Format) specification of the Ontolingua Frame syntax. It was found that many of the concepts need to be represented were exactly the same and that the lines between business and manufacturing were very hazy. As a result, PIF Project has been merged with the PSL (Process Specification Language) Project at NIST.

The Workflow Management Coalition has developed a Workflow Reference Model whose purpose is to identify the characteristics, terminology, and components to enable the development and interoperability of workflow specifications. Although the area of workflow is within the scope of the PSL project, it is only one small component The Workflow Reference Model has and will be used by the PSL project to ensure consistency.

In addition to the projects described above, there have been other, previous efforts aimed at creating process representations focusing specifically on various representational areas or on different functionality. For example, representational areas such as workflow, process planning, artificial intelligence planning, and business process re-engineering have had representations developed focusing solely on their respective areas. Equally important to the representational area in which the representations are being developed is the role (functionality) that the representation will play. There have been process representations developed which have focused on simply graphically documenting a process, to those which are used as internal representations for software packages, to those which are used as a neutral representation to enable integration. The process representations that resulted from many of these efforts were analysed in the second phase of the PSL project. A sampling of some of these existing process representations is shown in **Figure 4.1** and detailed information about the representations can be found in literature (Knutilla, A , et al , 1998)

4.5 Development of PSL

The PSL development approach involved five phases: requirements gathering, existing process representation analysis, language creation, pilot implementation and validation, and submission as a candidate standard.

The completion of the first phase resulted in a comprehensive set of requirements for specifying manufacturing processes (Schlenoff *et al.* 96) that were grouped into four major groups: the core, outer core, extensions and applications specific as described briefly below:

Core The most basic, essential requirements inherent to all processes. To represent process, it is either critical that these requirements be included, or these requirements are so common that every application either explicitly or implicitly uses them. While all processes contain core requirements, the core requirements provide the basis for representing only the simplest of processes, e g , resource, and task.

Outer Core: The pervasive, but not essential, requirements for describing processes common to most applications, e.g , temporal constraints, resource grouping, alternative tasks.

Extensions: The groupings of related requirements, common to some, but not all, applications that together provide an added functionality. Although the requirements listed within the extensions are not inherently necessary for representing processes, they are useful during implementation to provide their respective functionality. They are included here because the PSL must be able to represent information that will ultimately allow this functionality. The six extensions are Administrative/Business, Planning/Scheduling/Quality/Analysis, Real-Time/Dynamic, Process Intent, Aggregate Resources/Processes, and Stochastics/Statistics.

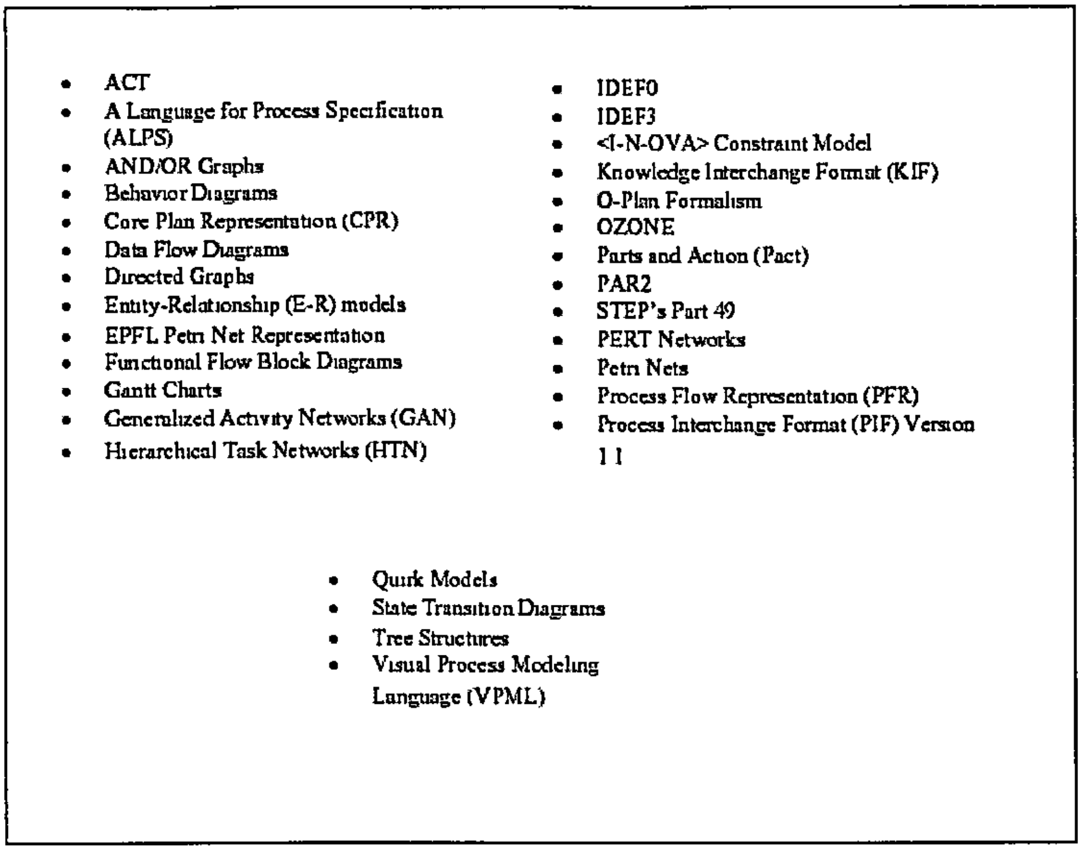
Application Specific: These are requirements only relevant within specific applications, e g., dynamic rescheduling for the production-scheduling environment.

These process specification requirements provided the context for analysing existing process representations.

In the second phase of the project, twenty-six process representations as shown in **figure 4.1** were analysed by the PSL team and analyzed with respect to the phase one to determine their applicability for representing the set of manufacturing process specification requirements identified and gathered in the first phase of the project (Knutilla, A., et al , 1998).

The objectives for analysing existing approaches for representing process included:

- Gain an improved understanding of existing approaches for representing process
- Identify how process specification requirements are represented within existing approaches
- Determine the strengths and limitations of existing approaches
- Identify the existing representations or combination of representations that provide the best coverage of all process specification requirements
- Understand and define what types of representations (e.g , object-oriented) provide the best coverage of all requirements
- Determine the completeness of process specification requirements identified in the first phase
- Refine the technical approach for developing a process specification language
- Identify the need as well as the candidates for PSL semantic concepts and their



- ACT
- A Language for Process Specification (ALPS)
- AND/OR Graphs
- Behavior Diagrams
- Core Plan Representation (CPR)
- Data Flow Diagrams
- Directed Graphs
- Entity-Relationship (E-R) models
- EPFL Petri Net Representation
- Functional Flow Block Diagrams
- Gantt Charts
- Generalized Activity Networks (GAN)
- Hierarchical Task Networks (HTN)
- IDEF0
- IDEF3
- <I-N-OVA> Constraint Model
- Knowledge Interchange Format (KIF)
- O-Plan Formalism
- OZONE
- Parts and Action (Pact)
- PAR2
- STEP's Part 49
- PERT Networks
- Petri Nets
- Process Flow Representation (PFR)
- Process Interchange Format (PIF) Version 1 1
- Quirk Models
- State Transition Diagrams
- Tree Structures
- Visual Process Modeling Language (VPML)

Figure 4.1 A Sampling of Existing Process Representations (source: Knutilla, A , et al , 1998)

4.5.1 PSL Implementations in Manufacturing Scenarios

Form the analysis of process representations it was concluded that nearly all of the representations studied focused on the syntax of process specification rather than the meaning of terms, the semantics. While this is sufficient for exchanging information between applications of the same type, such as process planning, different types of applications associate different meanings with similar or identical terms. As a result, the development of a formal semantic layer (an ontology) for PSL based on the Knowledge Interchange Format (KIF) specification (Genesereth, and Fikes, 1992) became the third important phase in the project. Based on the ontology developed to define explicitly and clearly the concepts intrinsic to manufacturing process information, PSL was used to integrate two manufacturing process applications: the ProCAP (KNO98) IDEF3 (MAY95) based process modelling and ILOG Scheduler 4 3 (ILO98) in the first pilot implementation of PSL (Schlenoff, et al. 1999) of the fourth phase of the project

4.5.2 PSL Extensions Development Approach

The development of PSL has proceeded on an as-needed basis. The initial PSL ontology was developed using a single scenario, the EDAPS (Electromechanical Design and Planning System) scenario developed by Steve Smith at the University of Maryland (Smith at al., 1996). The concepts introduced in that scenario were defined and modeled within PSL and later extended as other scenarios were explored. The PSL ontology was then further expanded to incorporate the concepts introduced in various manufacturing software applications when PSL was used to exchange process information among these packages As more software applications become "PSL-compliant," PSL will be continually expanded to ensure that all process-related concepts are capable of being represented within the language.

4.6 Semantics of Process Representations

In the second phase of PSL project (Knutilla, A., et al , 1998) from the analysis of twenty six process representations it was concluded that almost all existing approaches to process modelling lack an adequate specification of the semantics of the process terminology, which leads to inconsistent interpretations and uses of information. Hence, models tend to be unique to their applications and are rarely reused. Obstacles to interoperability arise from the fact that the systems that support the functions in many enterprises were created independently, and do not share the same semantics for the terminology of their process models. For example, **Figure 4.2** depicts (Schlenoff, et al. 2000) a situation in which two existing process-planning applications are attempting to exchange data. Intuitively the applications can share concepts; for example, both *material* in Application A and work-piece in Application B correspond to a common concept of *work-in-progress*. However, without explicit definitions for the terms, it is difficult to see how concepts in each application correspond to each other. Both Application A and B have the term *resource*, but in each application this term has a different meaning. Simply sharing terminology is insufficient to support interoperability; the applications must share their semantics, i.e., the *meaning* of their respective terminologies.

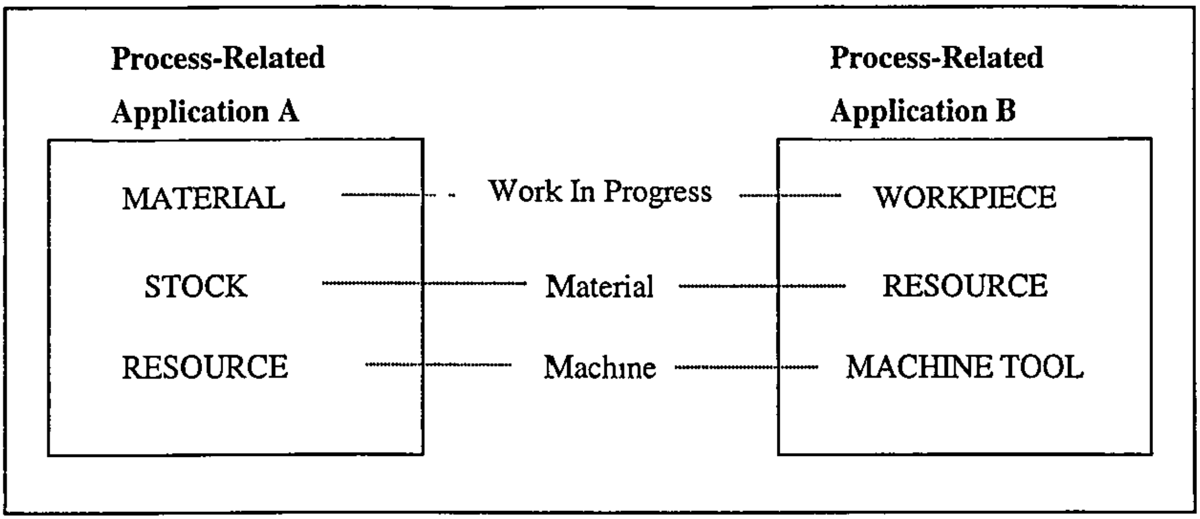


Figure 4 2: The Need for Semantics of Terminologies (source: Schlenoff, et al. 2000)

A ngorous foundation to the solution of such problem is the development of a formal specification of the semantics of process models. One approach to generating this specification is through the use of ontology. An ontology is a formal descnption of the entities within a given domain: the properties they possess, the relationships they stand in, the constraints they are subject to, and the patterns of behaviours they exhibit (Uschold, and Gruninger 1996). Ontology provides a common terminology that helps to capture key distinctions among concepts in different domains, which aids in the translation process. PSL was developed to address this issue by creating neutral standard language for specifying processes to integrate multiple process related applications.

4.7 The State of PSL: PSL Ontology

PSL is intended to include a default interchange language that manufactunng can use to exchange process information between process related applications. Currently, (http·//www mel.nist.gov/psl/) efforts have focused on developing the PSL data model that referred to as an ontology, which defines the meaning of the process-related terms in the language Therefore, as PSL stands now, it is more appropnate to refer to it as an ontology or a data model, which incorporates a syntax and grammar specification to make it a language.

An ontology is a lexicon of specialized terminology along with some specification of the meaning of terms in the lexicon. The pnmary component of PSL is ontology designed to represent the pnmitive concepts that, according to PSL, are adequate for descnbing basic manufacturing, engineering, and business processes and it is another aspect that makes PSL unique. All concepts in PSL are formally defined, using the Knowledge Interchange Format (KIF) (Genesereth, M , Fikes, R, 1992), to eliminate the ambiguity usually encountered when exchanging information among disparate applications. This ontology provides the backbone that enables and ensures correct translations. An arbitrary set of terms can be included in the ontology, but they can only be shared if there is an agreement on the meaning of the terminology. It is the intended semantics of the terms that is being shared, not simply the terms.

The challenge is that a framework is needed for making the meaning of the terminologies for ontologies explicit. Any intuitions that are implicit are a possible source of ambiguity and confusion. For the PSL ontology, it is necessary to provide a rigorous mathematical characterization of process information as well as precise expression of the basic logical properties of that information in the PSL language In providing ontology the following three notions specified:

- Language
- Model theory
- Proof theory

4.7.1 The Language

A language is a lexicon (a set of symbols) and a grammar that is a specification of how these symbols can be combined to make well-formed formulas or formal sentences. The lexicon consists of logical symbols (such as boolean connectives and quantifiers) and nonlogical symbols. For PSL, the nonlogical part of the lexicon consists of expressions (constants, function symbols, and predicates) chosen to represent the basic concepts in the PSL ontology. Notably, these will include the 1-place predicates 'activity', 'activity-occurrence', 'object', and 'timepoint' for the four primary kinds of entity in the basic PSL ontology, the function symbols beginof and endof that return the timepoints at which an activity begins and ends, respectively, and the 2-place predicates is-occurring-at, occurrence-of, exists-at, before, and participates-in, which express important relations between various elements of the ontology

The underlying grammar used for PSL is that of KIF (Knowledge Interchange Format). Briefly stated, KIF (Genesereth, M , Fikes, R, 1992) is a formal language based on first-order logic developed for the exchange of knowledge among different computer programs with disparate representations. KIF provides the level of rigor necessary to define concepts in the ontology unambiguously, a necessary characteristic to exchange manufacturing process information using the PSL Ontology.

4.7.2 Model Theory

The model theory of PSL provides a rigorous, abstract mathematical characterization of the semantics, or meaning, of the language of PSL an abstract representation of the primitive concepts of PSL. This representation is typically a set with some additional structure (e.g., a partial ordering, lattice, or vector space). The model theory then defines meanings for the terminology and a notion of truth for sentences of the language in terms of this model. Given a model theory, the underlying theory of the mathematical structures used in the theory then becomes available as a basis for reasoning about the concepts intended by the terms of the PSL language and their logical relationships, so that the set of models constitutes the formal semantics of the ontology.

4.7.3 Proof Theory

The proof theory of PSL is perhaps its most important component. It consists of three components: PSL Core, one or more foundational theories, and PSL extensions

PSL Core: the PSL Core is a set of axioms written in the basic language of PSL. The PSL core axioms provide a syntactic representation of the PSL model theory, in that they are sound and complete with regard to the model theory That is to say, every axiom is true in every model of the language of the theory, and every sentence of the language of PSL that is true in every model of PSL can be derived from the axioms. Because of this tight connection between the Core axioms and the model theory for PSL, the Core itself can be said to provide semantics for the terms in the PSL language.

Foundational Theories: the purpose of PSL Core is to axiomatize a set of intuitive semantic primitives that is adequate for describing basic processes. Consequently, its characterization of them does not make many assumptions about their nature beyond what is needed for describing those processes. The advantage of this is that the account of processes implicit in PSL core is relatively straightforward and uncontroversial. However, a corresponding liability is that the Core is rather weak in terms of pure logical strength. In particular, the theory is not strong enough to provide definitions of the many auxiliary notions that become needed to describe an increasingly broader range of processes in increasingly finer detail. (Auxiliary notions are axiomatized in PSL *extensions*) For this reason, PSL includes one or more *foundational theories.* A foundational theory is a theory whose expressive power is sufficient for giving precise definitions of, or axiomatizations for, the primitive concepts of PSL, thus greatly enhancing the precision of semantic translations between different schemes. Moreover, in a foundational, one can define a substantial number of auxiliary, and prove important meta-theoretical properties of the core and its extensions. There are several good foundational theories. Of these, set theory is the most familiar, and most powerful and its foundational capabilities are well known. For PSL's purposes, however, a more suitable foundation is a modified and extended variation of the situation calculus. The reason for this is that the situation calculus's own primitives situation, action, fluent (roughly, proposition) are already highly compatible with the primitives of PSL. It is very natural to identify PSL primitives with, or define them in terms of, the primitives of the situation calculus. In addition, the situation calculus is also strong enough to define a wide variety of auxiliary notions and, with the addition of some set theory; it can be used as a basis for proving basic metatheoretic results about the Core and its extensions as well.

Extensions· the final component of PSL consists of PSL *extensions.* PSL core is a relatively simple theory that is adequate for expressing a wide range of basic processes. However, more complex processes require expressive resources that exceed those of PSL core. PSL extensions give the resources to express information involving concepts that are not part of PSL core. Rather than clutter PSL core itself with every conceivable concept that might prove useful in describing one process or another, a variety of separate, modular extensions have been (and continue to be) developed that can be added to PSL core as needed. In this way a user can tailor PSL precisely to suit ones expressive needs

To define an extension, new constants and/or predicates are added to the basic PSL language, and, for each new linguistic item, one or more axioms are given that constrain its interpretation. In this way one provides "semantics" for the new linguistic items For example PSL core does not provide the resources to express information about time-durations. However, such a notion might be useful or even essential in many contexts. To facilitate the expressive needs for time duration, a theory of time durations has been developed which can be added to PSL core, thus providing the user with the desired expressive power.

When combined with a foundational theory like the situation calculus, a distinction can be drawn between *definitional* and *nondefinitional* extenstions. As the name suggests, a definitional extension is an extension whose new linguistic items can be completely defined in terms of the foundational theory and PSL core. Theoretically, then, definitional extensions add no new expressive power to PSL core or foundational theory. However, because definitions of many subtle notions can be quite involved, definitional extensions can prove extremely useful for describing complex processes in as succinct a manner as possible. Nondefinitional extensions, on the other hand are extensions that involve at least one notion that cannot be defined in terms of PSL core and the chosen foundational theory.

4.8 Consistency and Completeness of the PSL Ontology

One of the advantages of using a formal approach to ontology design is that the axioms and definitions of the ontology are consistent, and that they are complete with respect to the intended models, which are specified in the model theory of the ontology. It is shown that the axioms in a PSL extension are consistent by constructing a model that satisfies all of the axioms. Then a complete characterization of these models also provided by showing that any structure that satisfies the axioms in the extension is one of the intended models.

4.9 PSL Architecture

The three components of the PSL architecture and their relations are illustrated in **Figure 4.3.** The solid arrows indicate the definability relation. The dashed lines indicate partial definability, i e , the case where some, but not all the additional linguistic items in the language of an extension are definable. Two or more solid arrows pointing to the same oval indicate the possibility that more than one given theory might jointly be used to define a new extension. PSL core is connected to foundational theories, but this would not sufficiently distinguish the central role of the Core from the more auxiliary roles of extensions. Hence, PSL Core is pictured as sitting directly upon the foundational theories. "(PSL Core and Foundation Theory) " in the PSL Core box indicates that PSL Core together with a foundational theory are used to formulate definitional extensions.

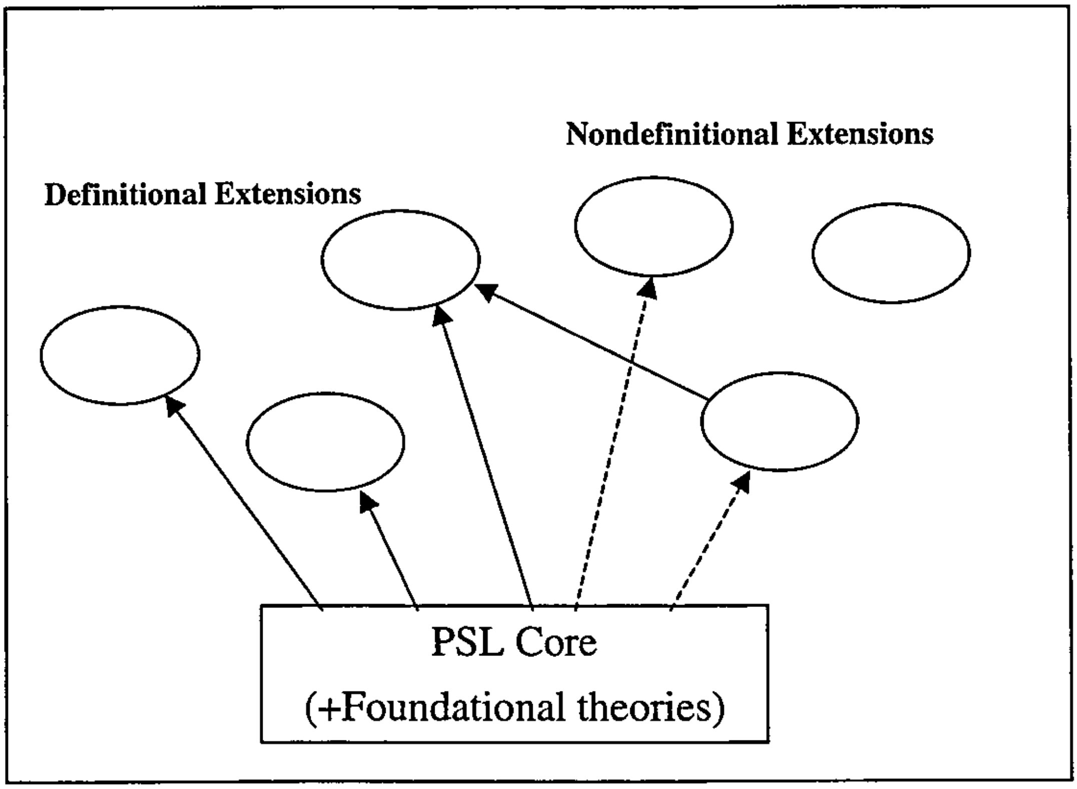


Figure 4 3. The PSL Semantic Architecture (source: Schlenoff, et al. 2000)

3.9.1 Informal Semantics of PSL-Core

PSL-Core is based upon a precise, mathematical, first-order theory, i.e., a formal language, a precise mathematical semantics for the language, and a set of axioms that express the semantics in the language. Here we will provide a brief informal sketch of the semantics. There are four primitive classes, two primitive functions, and three primitive relations in the ontology of PSL-Core The classes are OBJECT, ACTIVITY, ACTIVITY_OCCURRENCE and TIMEPOINT. The four relations are PARTICIPATES-IN, BEFORE, and OCCURRENCE-OF. The two functions are BEGINOF, and ENDOF. ACTIVITIES, ACTIVITY_OCCURRENCES, TIMEPOINTs (or "POINTs", for short), and OBJECTs are known collectively as entities, or things. These classes are all pairwise disjointed.

Intuitively, an OBJECT is a concrete or abstract thing that can participate in an ACTIVITY. The most typical examples of OBJECTs are ordinary, tangible things, such as people, chairs, car bodies, NC-machines, though abstract objects, such as numbers, are not excluded. OBJECTs can come into existence (e g., be created) and go out of existence (e.g., be "used up" as a resource) at certain points in time. In such cases, an OBJECT has a begin and/or end point. Some OBJECTs, e.g., numbers, do not have finite begin and end points. In some contexts it may be useful to model certain ordinary OBJECTs as having no such points either.

An ACTIVITY-OCCURRENCE is a limited, temporally extended piece of the world, such as the first mountain stage of the 1997 Tour de France or the eruption of Mt. St. Helen. Any ACTIVITY-OCCURRENCE is simply taken to be characterized chiefly by two things: its temporal extent, as determined by it's begin and end POINTs (possibly at infinity), and the set of OBJECTs that participate in that ACTIVITY at some point between its begin and end POINTs.

TIMEPOINTs are ordered by the BEFORE relation. This relation is transitive, non-reflexive, total ordering. In PSL-Core, that time is not dense (i.e , between any two distinct TIMEPOINTs there is a third TIMEPOINT), though it is assumed that time is infinite. POINTs at infinity (INF+ and INF-) are assumed for convenience. (Denseness, of course, could easily be added by a user as an additional postulate.) Time intervals are not included among the primitives of PSL-Core, as intervals can be defined with respect to TIMEPOINTs and ACTIVITIES. TIMEDURATIONS are included in an extension of the PSL-Core that builds upon (Hayes, P., 1996).

The basic notions of the PSL-Core are axiomatized formally as a first-order theory. These axioms simply capture, in a precise way, the basic properties of the PSL ontology.

4.10 Requirements for PSL extensions

A goal of PSL (ISO DIS 18629-1,) is to facilitate applications interoperability by means of the development of translators between the native formats of those applications and PSL. To support this goal, each part of the standard will be composed of one or more extensions to PSL-Core. For each extension, the standard will include the following:

- Non-logical lexicon
- Specification of models
- Set of axioms
- Theorems for the verification of extension

- Grammar for process descriptions that use the terminology of the non-logical lexicon.

4.10.1 Non-logical Lexicon

The non-logical lexicon is the terminology of the standard that corresponds to the concepts and relationships related to manufacturing processes. All terms in the non-logical lexicon of the standard are constant, function, or relation symbols in KIF. Each extension specifies a unique non-logical lexicon. Any term in the standard shall belong to the non-logical lexicon of a unique extension.

4.10.2 Specification of Models

The specification of model for PSL provides a rigorous abstract mathematical characterisation of the semantics of the terminology of PSL. This characterisation defines the meanings of terms with respect to some mathematical structures together with a notion of truth with respect to those structures for sentences of the language.

4.10.3 Axioms of the extensions

The axioms of PSL are the set of KIF sentences that constrain the interpretation of the terminology of the non-logical lexicon The axioms are organised into PSL-Core and a partially ordered set of extensions to PSL Core. An extension provides the logical expressiveness to express information involving concepts that are not explicitly specified in PSL-Core. All extensions within PSL shall be consistent extensions of PSL-Core, and may be consistent extensions of other PSL extensions. However, not all extensions within PSL need to be mutually consistent.

4.10.4 Grammar for Process descriptions

The underlying grammar used for PSL is that of KIF (Knowledge Interchange Format). KIF is a formal language based on first-order logic developed for the exchange of knowledge among different computer programs with disparate representations. KIF provides the level of rigour necessary to unambiguously define concepts in the ontology. Process descriptions are sentences in KIF that use the non-logical lexicon of PSL. In particular, process descriptions will be restricted to sentences that are satisfied by elements in a model of the axioms of PSL. Process descriptions are not arbitrary sentences. Each extension shall have an associated BNF grammar for process descriptions, which extends the BNF grammar associated with PSL-Core.

4.10.5 Format for Extensions

All extensions have the following header information:

- Extension name: Core theory names with a suffix of the form .th, and definitional extension names with a suffix of the form def.
- Primitive lexicon,
- Defined lexicon;
- Core theories required by the extension. This is a list of PSL-Core theories, each of which is an extension of PSL-Core The given extension is an extension of the set of axioms which is the union of all theories in the list, together with the axioms in the extension;
- Definitional extensions required by the extension: This is a list of definitional extensions, each of which is an extension of PSL-Core

The content of any PSL extension is a set of KIF sentences. For core theories, there is a set of arbitrary KIF sentences for the primitive lexicon of the set of axioms. For definitional extensions and terms in the defined lexicon of a Core theory, each term shall have a conservative definition. In addition, each KIF sentence must have corresponding text in English that summarises the key intuitions captured by the sentence.

4.11 ISO 18629 PSL Standards

PSL is being standardized within Joint Working Group 8 of Sub-committee 4 (Industrial data) and Sub-committee 5 (Manufacturing integration) of Technical committee ISO TC 184 (Industrial automation systems and integration) as ISO 18629. More information about PSL standardization and on the evolving PSL Standard is available from http //www.mel nist.gov/psl/. The following sections present the parts and series of the ISO 18629 PSL standard and their organizations

4.11.1 Organization of the ISO 18629 PSL Standard

The components of the ISO 18629 PSL standard are grouped into the following parts: Note that not all of them are completely developed to date.

a) Part 1: Overview and basic principles

There are two types of extensions within ISO 18629: they are the core theories and the definitional extensions.

b) Part 1x series: Core theories

The current contents of these series of parts addresses:

Part 11: PSL-Core,

Part 12: Outer Core,

Part 13: Duration and ordering theories;

Part 14: Resource theories;

Part 15: Actor and agent theories.

Any new core theory will have to be included in the Part 1x series

c) Part 2x series: External mappings.

The current expected content of this series of part includes:

Part 21: EXPRESS;

Part 22: XML;

Part 23: UML.

This set of mappings may evolve according to industry needs and technology changes.

d) Part 4x series: Definitional extensions

In addition to the core theories, ISO 18629 PSL provides a series of definitional extensions used to capture the semantics of process terminology in different applications. All definitions in these extensions use the terminology of the core theories. The series currently includes:

Part 41: Activities;

Part 42: Temporal and state;

Part 43: Activity ordering and duration;

Part 44: Resource roles;

Part 45: Resource sets;

Part 46: Processor activities,

Part 47: Process intent

Additional extensions are to be developed later according to industry needs by any standardisation committee. Any new extension that is a definitional extension of PSL-Core will be included in the Part 4x series.

e) Part 2xx series: Translator Implementation Guidelines

Parts of this series will be developed according to industry needs and technology changes.

4.12 ISO 18629-1x series Core theories

4.12.1 ISO 18629-11 PSL-Core

The PSL-Core is based upon a precise, mathematical, first-order theory that is a formal language, a precise mathematical semantics for the language, and a set of axioms, which express the semantics in the language. The basic elements of the language are four primitive classes, two primitive functions, and seven primitive relations of the ontology of the PSL-Core

- The primitive classes are activity, activity_occurrence , timepoint, and object.
- The two functions are `beginof` and `endof`.
- The seven relations are `before, occurrence_of, between, before-eq, between-eq, is-occurring-at, participates-in, exist-at`.

4.12.2 ISO 18629-12 Outer Core

This set of extensions of PSL-Core is used in defining those extensions that shall be used in practice for specifying manufacturing processes that belong to individual applications. The set of extensions in ISO 18629-12 augment PSL-core but they are not expressive enough to specify the complex items found in practice. The extensions in ISO 18629-12 are more generic and pervasive in their applicability than the rest of the extensions except PSL-Core in ISO 18629. The extensions in ISO 18629-12 are less generic than the PSL-core. The main difference is that

PSL-Outer Core requires PSL-Core for its specifications whereas PSL-Core does not require any other set of axioms for its specifications These extensions are:

- Occurrence Trees;
- Discrete States;
- Subactivity;
- Atomic Activity;
- Complex Activity;
- Activity Occurrence.

The Occurrence Tree extension introduces a tree structure over the set of possible activity occurrences; branches in the tree correspond to different sequences of primitive activity occurrences.

The Discrete State extension specifies the basic concepts for states and their relationships to activity occurrences. In particular, all discrete states are changed by activity occurrences, but they do not change during an activity occurrence.

The Subactivity extension describes how activities can be aggregated and decomposed

The Atomic Activity extension introduces the class of concurrent activities.

The Complex Activity extension specifies the relationship between occurrences of the subactivities of an activity and occurrences of the activity itself.

The Activity Occurrence extension defines relations that allow the description of how activity-occurrences relate to one another with respect to the time at which they start and end

Figure 4.4 shows (ISO DIS 18629-12,) the relationships between these theories within the Outer Core, in which the arrows represent the dependencies among the theories. All theories in the Outer Core are extensions of PSL-Core. The Atomic Activity Core theory is an extension of both the Subactivity and the Occurrence Trees theories, while the Discrete States Core theory is an extension of the Occurrence Trees core theory alone. The Activity Occurrence Core theory is an extension of the Complex Activities core theory, which in turn is an extension of Atomic Activities.

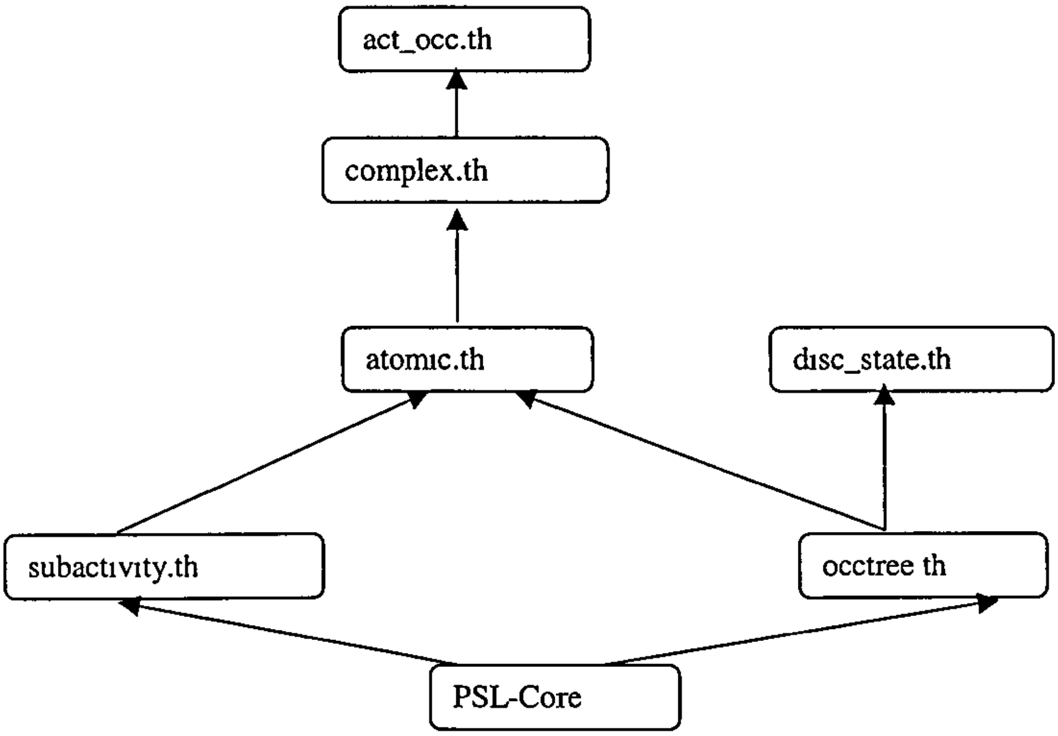


Figure 4.4: Relationships among sets of axioms within PSL Outer Core.

These extensions, together with PSL-Core, provide much of the infrastructure for specifying the definitions of terminology within ISO 18629. Every extension within the current version of ISO 18629 is an extension of one or more of these theories.

The following sets of theories define other extensions of the Outer Core, required by some, though not all, definitional extensions within ISO 18629.

4.12.3 ISO 18628-13 Time and ordering theories

The extensions related to time and ordering theories are:

- Duration;
- Subactivity occurrence ordering
- Iterated activities
- Occurrence tree automorphisms
- Envelopes and umbrae.

The duration extension introduces the concept of duration as a relationship among timepoint. This allows the introduction of quantitative concepts related to time, as well as providing the basis for defining the duration of activities, activity occurrences, and objects.Intuitively, duration is the "difference" between two timepoint in the timeline

The subactivity occurrence ordering extension specifies the relations required to represent various kinds of partially ordered sets of activities. This includes sequences, parallelism, AND splits/junctions, and OR splits/junctions.

4.12.4 ISO 18629-14 Resource theories

The extensions related to resource theories are:

- Resource requirements,
- Resource sets.

The resource requirements extension turns into axioms of the concept of resource as any object, which is required by an activity. In particular, resources are defined with respect to the possible interactions among activities.

The resource set extension turns into axioms of the concept of a set of resources, which as a whole also satisfy the axioms for resources. Different kinds of resource sets are defined in Part 44, and they include such concepts as resource pools (sets of machines) and buffers (sets of inventory resources).

4.12.5 ISO 18629-15 Activity performance theories

The extension related to activity performance theories is:

- Activity performance

The Activity performance extension turns into axioms the relationship that holds between an activity and the actor (such as a human or machine) who performs the activity.

4.13 ISO 18629-2x series External mappings

In addition to specifying the grammar for process descriptions, ISO 18629 also specifies mappings between this grammar and languages used by other manufacturing standards. In particular, there shall be mappings between the grammar of ISO 18629 and EXPRESS (to facilitate interoperability with applications using ISO 10303), as well as mappings to XML and UML.

These other languages are used as alternative ways of representing particular process descriptions, rather than in the specification of the semantics of the terminology of ISO 18629.

4.14 ISO 18629-4x series Definitional Extensions

4.14.1 ISO 18629-41: Activity extension

The fundamental theories that are part of ISO 18629-41 are (ISO CD 18629-41,):

- Deterministic Activities: Permuting Branch Structure;
- Non-deterministic Activities: Folding Branch Structure;
- Non-deterministic Activities: Branch Structure and Ordering;
- Non-deterministic Activities: Repetitive Branch Structure;
- Spectrum of Activities: Permuting Activity Trees;
- Spectrum of Activities: Compacting Branch Structure;
- Spectrum of Activities: Activity Trees and Re-ordering;
- Spectrum and Sub-tree Containment;
- Embedding Constraints for Activities;
- Skeletal Activity Trees;
- Atomic Activities: Upwards Concurrency;
- Atomic Activities: Downwards Concurrency;
- Spectrum for Atomic Activities;

- Preconditions for Activities.

Figure 5 shows the relationships between these extensions and the foundational theories of ISO 18629-12 that are useful for specifying definitional extensions in ISO 18629-41.

The arrows show dependencies between extensions in ISO 18629-41 and ISO 18629-12. All theories in ISO 18629-41 are extensions of the ISO 18629-11, itself an extension of the ISO 18629-12. Only the parts of ISO 18629-12 that are useful to showing dependencies with extensions in ISO 18629-41 are shown in **Figure 4.5.**

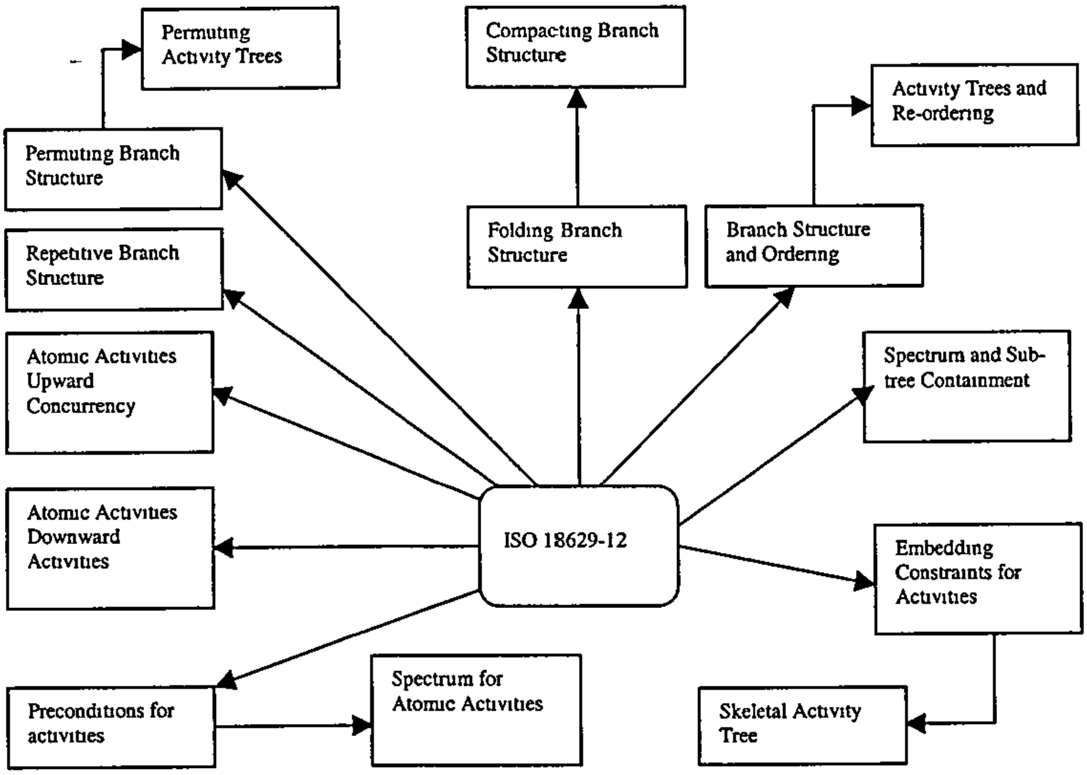

Permuting Activity Trees
Compacting Branch Structure
Activity Trees and Re-ordering
Permuting Branch Structure
Repetitive Branch Structure
Folding Branch Structure
Branch Structure and Ordering
Atomic Activities Upward Concurrency
Spectrum and Sub-tree Containment
Atomic Activities Downward Activities
ISO 18629-12
Embedding Constraints for Activities
Preconditions for activities
Spectrum for Atomic Activities
Skeletal Activity Tree

Figure 4.5. Definitional Extensions of ISO 18629-41

4.14.2 ISO 18629-42: Temporal and State extension

The extensions in this part are defined with respect to the Part 12 Outer Core and the Part 13 Time and ordering theories. They include:

- Preconditions for Activities
- State-based Preconditions for Activities

- Time-based Preconditions for Activities
- Preconditions based on State and Time
- Occurrence-based Preconditions for Activities
- Preventable Preconditions for Activities
- Periodic Preconditions for Activities
- Spoilage Preconditions for Activities
- Effects of Activities
- State-based Effects of Activities
- Time-based Effects of Activities
- Occurrence-based Effects of Activities
- Effects of Activities: State and Occurrences
- Effects of Activities: Time and Occurrences
- Fluent Trees
- Distribution of Complex Activities
- State-based Distribution of Complex Activities
- Time-based Distribution of Complex Activities
- Distribution based on State and Time
- Variation of Complex Activities
- State-based Variation of Complex Activities
- Time-based Variation of Complex Activities
- Variation based on State and Time
- Variation of Atomic Activities: Upward Concurrency
- Variation of Atomic Activities: Downward Concurrency
- Interfering Preconditions
- Clobbering Effects
- Variation of Interfering Preconditions
- Variation of Clobbering Effects

The interval activities extension considers kinds of activities that can be defined using the relations in the state constraints extension In particular, it considers activities that have the property of being interruptible or non-interruptible.

The state constraints extension defines relations between states and activity occurrences. These include concepts such as preconditions and effects, as well as the notions of an activity achieving or falsifying various states.

4.14.3 ISO 18629-43: Activity ordering and duration extension

The extensions in this part are defined with respect to the Part 12 Outer Core and the Part 13 Time and ordering theories. These include.

- Strong Partially Ordered Activities
- Weak Partially Ordered Activities
- Duration Constraints for Activity Occurrences)
- State-based Duration
- Time-based Duration
- Duration based on State and Time
- Ordering and Duration Constraints on Activity Occurrences
- Ordering and Duration Constraints on Embedded Activity Occurrences
- Spoilage Preconditions for Activities
- Scheduled Embedding Constraints
- Interruptable Activities
- Duration-based Effects
- Effects of Activities based on Duration and State
- Classes of Iterated Activity Occurrences

The complex sequence ordering relations extension is restricted to non-deterministic activities. It supports the specification of activities in which there is branching and conditional occurrence of sub-activities.

The ordering relations over activities extension are restricted to deterministic activities. It specifies different kinds of activities in which there is a partial ordering over sub-activity occurrences.

The temporal ordering constraints extension specifies relations among the occurrences of activities with respect to the times at which they occur. These relations are particularly used within scheduling applications.

4.14.4 ISO 18629-44 Resource roles

The extensions in this part are defined with respect to the Part 12 Outer Core and the Part 14 Resource theories. They include:

- Capacity-based concurrency;
- Reasoning about resource divisibility;
- Resource role;
- Reasoning about resource usage.

The capacity-based concurrency extension defines different kinds of activities and resources using concurrency constraints.

The resource divisibility extension considers the different ways in which resources can be shared by multiple activities.

The resource roles extension defines various roles that resources play with respect to activities; these include reusable, consumable, and renewable.

The resource usage extension characterises constraints on resources over the intervals in which activities occur.

4.14.5 ISO 18629-45 Resource sets

The extensions in this part are defined with respect to the Part 12 Outer Core and the Part 14 Resource theories. These include:

- Homogeneous resource sets
- Inventory resource sets
- Resource pools
- Resource set-based activities
- Substitutable resources

The primary purpose of these extensions is the set of axioms for discrete capacity resources for which the discreteness of the resource arises from the fact that it is actually composed of a set of resources, and any activity requires or provides some subset of resources in this set.

Homogeneous sets define different kinds of substitutable resources
Inventory resource sets are related to buffers

Resource pools are equivalent to discrete capacity resources.

Resource set-based activities define kinds of activities that use resource sets.
Substitutable resources make the distinction between sets of arbitrary resources and sets of resources that can be substituted for others in an activity.

4.14.6 ISO 18629-46 Processor Activities

The extensions in this part consider a particular kind of activities that are defined with respect to the roles of the resources required by the activities This includes:

- Processor activities
- Resource paths

Processor actions are actions that use some set of resources, consume some set of material resources, and produce or modify some other set of material resources.

Resource paths are partially ordered sets of processor actions in which the output material resource of one processor action is the input material of the next processor action.

4.14.7 ISO 18629-47 Process intent

This number has been reserved for the development of appropriate extensions.

4.15 ISO 18629-2xx series Translator implementation guidelines

The guidelines in the 200 series of ISO 18629 identify the different extensions within the ontology that are necessary for developing translators among applications in different manufacturing domains. Possible examples of use are:

- Process modelling
- Process planning

- Production planning
- Project management
- Scheduling
- Simulation
- Process execution

4.16 Summary

The Process Specification Language (PSL,) was developed, at National Institute of Standards and Technology (NIST), as a neutral language for manufacturing processes specification and to address the issue of interoperability problems among process related manufacturing software applications. PSL has been standardized within Joint Working Group 8 of Sub-committee 4 (Industrial data) and Sub-committee 5 (Manufacturing integration) of Technical committee ISO TC 184 (Industrial automation systems and integration) as *ISO 18629.*

The development of PSL has proceeded on an as-needed basis. The initial PSL ontology was developed using a single scenario and later extended as other scenarios were explored and when PSL was used to exchange process information among the packages in the scenarios. After the initial development of PSL in the EDAPS (Electromechanical Design and Planning System) scenario PSL was extended to incorporate concepts introduced in the IDEF3-based ProCAP process modelling tool and the C++ based ILOG Scheduler when PSL was used to exchange process information between these two packages in the first pilot implementation of the language in the manufacturing environment. The PSL ontology is still under development and new concepts are still appearing in the ontology, however there were no other implementations and there is no clear indication about what these new concepts relate to and their target applications for interoperability.

Although PSL was mainly developed in the manufacturing environment for process related manufacturing applications interoperability, as a first research on PSL in the construction environment, the applicability of the Process Specification Language (PSL) in construction has been studied at Stanford University focusing on Vite™ , which is a project and organization modeling system designed to assist in developing organizational structures and identifying potential problems with project cost, time, or quality. (Gruninger, et al 2003). This was to analyse this emerging standard Process Specification Language and research the possibility for adapting this standard to the problem area of software applications' interoperability in the construction environment.

Chapter 5 Research Methodology and Design

5.1 Literature Review and Theoretical Framework

The first section of this chapter discuses selected elements of the literature review on research approaches and methodologies and outlines the research strategy taken for this study. This is followed by, sections on research methods and approaches used.

5.1.1 Research Approaches in Construction IT

Research in construction information technology (IT) is not the study of pure technological problems alone, concerns are closely related to human activities, organisational as well as technological issues. Construction IT research constitutes complex phenomena with the need for appropriate research methods for producing an understanding of the complex organisational context of the IT applications, the process whereby the IT applications influence and are influenced by the context, and for addressing the problems of the sector. However, researchers, rather than adopting research approaches appropriate to the research in question, tend to unquestioningly inherit the positivist paradigm, which is the dominant approach in construction management research by the majority of researchers (Seymour et al. 1997). Therefore, there is a need for transition from single and/or inherited approaches to an appropriate research method (whether single or mixed research methodology). While Information Systems (IS) research (Myers, Michael D. 1999; Avison et al., 1993; Lee et al., 1997; Ngwenyama et al , 1999, Nissen et al., 1991) has seen shifts from pure technological to managerial and organisational issues, the majority of the researches work in construction IT is still focusing on pure technological issues in which the organisational context of the IT applications is ignored or not taken into account. As a result there have been sophisticated technological developments however the impact on construction practices have not been successful (Leilie, 1997; Wix, 1998; and Eastman, 1997).

The evolution of Research in construction management (Carter and Fortune 2002) has been the focus of debate for construction researchers in the industry. The focus of research in construction has evolved from on site operations aimed at improving construction processes in the 1960, to integration of operation, and with the IT revolution of the 1980s that brought multitude of construction software, the emphasis of research moved to construction processes (McCaffer and Edum Fotwe, 1999). The rapid cultural and technological changes in the industry have changed the requirements of research, and the need for different approaches has been recognised to addressing the problems of the industry (Edum Fotwe et al. 1996). And now with a wide use of computer applications in the industry, the focus of research has shifted to more or less pure technological issues. Research in construction has been continuously changing to hit the target of addressing a specific problem of the industry at a time. However, the philosophical assumptions that guide construction research and the use of methods of research have not seen much change. The complex organisational context of IT applications in construction needs complex research methods that are multi-paradigm approaches if the problem, which are interwoven in the organisation of the industry is to be addressed.

There are unavoidable relations between the construction operations, IT tools and applications and human activities, who carry out the operations using the IT applications. However, the focus of this research (research in construction IT) is not seen as how the construction operations, IT applications and human activities are related organised and work together, or how the human activities interact with IT applications, but how the humans communicate and interact with each other using IT applications. This brings into consideration issues from computer science (IS and IT) research. However, the construction processes, humans or professional participants, IT applications including their organisational issues require studying through literature review and some kind of observation, hence the need for issues to be considered from sociology and psychology as well as computer science. Therefore, the development of theoretical framework for this research study has been influenced by the discussions and literature on paradigms (at the philosophical and technical level) in the computer science and within the new discipline of Information Systems (IS) (Avison et al., 1993; Lee et al., 1997, Ngwenyama et al., 1999; Nissen et al., 1991; Fitzgerald & Howcorft, 1998; and Walsham, 1993), as well as the discussion of research methodologies in construction (Carter. K, and Fortune C. 2002) and manufacturing (Schlenoff, C , Knutilla, A , Ray, S., 1998, 1997; Schlenoff, C., Gruninger M , Tissot, F., Valois, J , Lubell, J , Lee, J.,2000) researches.

5.1.2 Philosophical Basis of Research

Research is based on some underlying assumptions about what constitutes 'valid' research and which research methods are appropriate. In order to conduct and/or evaluate research, it is therefore important to be aware of these assumptions that guide a research. These are referred to as philosophical assumptions of research approaches. Philosophical assumptions for a research can be related to the underlying ontology, which refers to the study of being, and about the state/nature of the world (what exists?), and our presuppositions about the nature of 'reality', 'being', etc.; epistemology, which is the assumptions about knowledge and how it can be obtained (the philosophical theory of knowledge); methodology, the system of methods followed in a particular discipline or the branch of philosophy that analyses the principles and procedures of inquiry in a particular discipline; and axiology, the study of values and value judgments (Guba, E., & Lincoln, Y.1994 and Fitzgerald and Howcroft 1998). It can be seen from **Figure 5.1**, that there are soft and hard positions at different level of abstractions that guide research approaches These are the main dichotomies characteristics of each research tradition categorized according to various levels, namely, ontological, epistemological, methodological, and axiological. These paradigms provide a theoretical and methodological framework from which to view and make sense of a research subject. Thus a research subject is contrasted in the five sets of assumptions at the five levels shown in figure 1., these assumptions move gradually to lower levels of positions and influence a research process.

At the **methodological level** of **figure 5.1,** research approaches can be classified in various ways, however one of the most common distinctions is between qualitative and quantitative research methods. The two major research approaches, the qualitative and quantitative are seen as soft and hard approaches. The quantitative research approach is seen as objective that is relating to a phenomenon or conditions independent of the individual thought and perceptible to all observers, and relying heavily on statistics (Lee, 1992). On the other hand the qualitative approach is seen as subjective, relating to experience or knowledge as conditioned by personal mental characteristics or states and preferring language and description. These distinctions are useful in recognizing the two approaches however; each of the approaches can be guided by differing paradigms.

At the **ontological level**, to the realist the social world (the social world is real and external to our individual descriptions) is tangible, made up of relatively immutable structures that exist independently of our individual descriptions. The relativist however views reality as created in concepts that are used to structure that reality hence there are multiple realities Critical realism is a modern realism with a critical twist: Critical realists believe that reality is affirmed, but the way that reality is represented/experienced/interpreted is seen to be shaped by social, culture, language, political interests, race, gender, class etc. The unsuccessfulness of research informed by positions at the extreme of continuum between hard and soft research approaches and the need for mutual integration of the realist and relativist positions has been explored by (Fitzgerald and Howcroft 1998).

At the **epistemology level**, while Positivists generally assume that reality is objectively given and can be described by measurable properties, which are independent of the observer (researcher), the Interpretivists relies on human interpretations of reality as the basis of understanding the world Rather than focusing on the objective reality interpretivist are more interested in how people interpret that reality and how they act based on those interpretations (Creswell 1994) While positivist generally attempt to test theory, in an attempt to increase the predictive understanding of phenomena (Orlikowski and Baroudi 1991), Interpretivists understand phenomena through the meanings that people assign (Walsham 1993). The objective versus subjective research methods (Burrell and Morgan, 1979), are concerned with the discovery of general laws (nomothetic) versus being concerned with the uniqueness of each particular situation (idiographic), as aimed at prediction and control versus aimed at explanation and understanding, as taking an outsider (etic) versus taking an insider (emic) perspective, and so on. Axiological assumptions are closely related to the epistemological. These assumptions are regarding the role of values. The rigor and relevance at the Axiological level are highly valued by researchers in construction and there is a debate concerning how to achieve these (Crook, et al.1996, Fenn, 1997, Seymour, et al., 1996, Seymour and Rooke, 1998, Runeson, 1997, and Walker, 1997).

SOFT	HARD
ONTOLOGICAL LEVEL	
Relativist Belief that multiple realities exist as subjective constructions of the mind Socially-transmitted terms direct how reality is perceived and this will vary across different languages and cultures	**Realist** Belief that external world consists of pre-existing hard, tangible structures which exist independently of an individual's cognition
EPISTEMOLOGICAL	
Interpretivist No universal truth Understand and interpret from researcher's own frame of reference Uncommitted neutrality impossible Realism of context important	**Positivist** Belief that world conforms to fixed laws of causation Complexity can be tackled by reductionism. Emphasis on objectivity, measurement and repeatability
Subjectivist Distinction between the researcher and research situation is collapsed Research findings emerge from the interaction between researcher and research situation, and the values and beliefs of the researcher are central mediators	**Objectivist** Both possible and essential that the researcher remain detached from the research situation Neutral observation of reality must take place in the absence of any contaminating values or biases on the part of the researcher
Emic/Insider/Subjective Origins in anthropology Research orientation centered on native/ insider's view, with the latter viewed as the best judge of adequacy of research	Etic/Outsider/Objective Origins in anthropology Research orientation of outside researcher who is seen as objective and the appropriate analyst of research
METHODOLOGICAL LEVEL	
Qualitative Determining what things exist rather than how many there are Thick description Less structured and more responsive to needs and nature of research situation	**Quantitative** Use of mathematical and statistical techniques to identify facts and causal relationships Samples can be larger and more representative Results can be generalized to larger populations within known limits of error
Exploratory Concerned with discovering patterns in research data, and to explain/understand them Lays basic descriptive foundation May lead to *generation* of hypotheses	**Confirmatory** Concerned with hypothesis testing and theory verification Tends to follow positivist, quantitative modes of research
Induction Begins with specific instances which are used to arrive at overall generalizations which can be expected on the balance of probability New evidence may cause conclusions to be revised Criticized by many philosophers of science, but plays an important role in theory/hypothesis conception	**Deduction** Uses general results to ascribe properties to specific instances An argument is valid if it is impossible for the conclusions to be false if the premises are true Associated with theory verification/falsification and hypothesis testing
Field Emphasis on realism of context in natural situation, but precision in control of variables and behavior measurement cannot be achieved	**Laboratory** Precise measurement and control of variables, but at expense of naturalness of situation, since real-world intensity and variation may not be achievable.
Idiographic Individual-centered perspective which uses naturalistic contexts and qualitative methods to recognize unique experience of the subject	**Nomothetic** Group-centered perspective using controlled environments and quantitative methods to establish general laws
AXIOLOGICAL LEVEL	
Relevance External validity of actual research question and its relevance to practice vital, rather than constraining the focus to that researchable by "rigorous" methods	**Rigor** Research characterized by hypothetico-deductive testing according to the positivist paradigm, with emphasis on internal validity through tight experimental control and quantitative techniques

Table 5.1. Summary of Soft vs Hard Research Dichotomies (adopted from: Fitzgerald and Howcroft 1998)

5.1.3 Critical Research

The main task of critical research is seen as being one of social critique, whereby the restrictive and alienating conditions of the status quo are brought to light. Critical research focuses on the oppositions, conflicts and contradictions in contemporary society, and seeks to be emancipatory i.e. it should help to eliminate the causes of alienation and domination The critical paradigm is similar to the interpretive, but goes a step beyond it in that it is very value-oriented. This paradigm takes a critical look at society and tries to identify inequities as well as ways to remedy them. The critical theory paradigm has been cast as combining an essentially realist ontology with a subjectivist epistemology (Guba 1990),

5.1.4 Triangulation or Pluralistic Research Approaches

The use of mixed methodologies or the positioning of a research at two or more extremes of positions is called triangulation or pluralistic approach. This would allow for a contingent tool-box approach where different methods with complementary strengths could be used as appropriate (Landry and Banville 1992; McGrath 1984).

The research positions on both side of table 5.1 , have strengths and weakness or limitations. However, it is argued that (Morgan 1983), it is wrong to condemn any research perspectives, reject a position of knowledge or replace one approach with another but adopting a multi paradigm strategy is the best alternative. In the literature these research approaches/ paradigms have been viewed as mutually exclusive opposites, (Burrell and Morgan 1979). Guba and Lincoln (1994) stated that interpretivism and positivism "cannot be logically accommodated anymore than ideas of contradiction. However, some researchers have argued that these positions should not be viewed as mutually exclusive (e g , Firestone 1990; Gable 1994; Iivari 1991; Jick 1983; Morey & Luthans 1984; and Patton 1990). For example, qualitative techniques complemented quantitative ones to interpret empirically determined statistical relationships in a successful research study (Kaplan and Duchon 1988). Thus, methodological pluralism has been recommended as an appropriate strategy in practice (Landry and Banville 1992; Iivari 1991; Nissen, Klein and Hirschheim 1991). Additionally, there is an argument that qualitative and quantitative research methods are not mutually exclusive, and it should be possible to combine in order to tackle the diversity of construction research (Raftery et al. 1997).

5.1.5 Basic and Applied Research

Newman (1994) and Galliers (1991) in their classifications of research methods draw a distinction between basic and applied research. Basic research involves "theory building" and "testing" and contributes to the generalization of knowledge To a certain extent, this kind of research can only be conducted after a field of study has reached a certain level of maturity and has all the parameters clearly defined to be generalized in a form of an appropriate theory: an established paradigm (Kuhn, 1970). Applied research, on the other hand, targets a specific problem relating to practice. The result of such research is intended to help practitioners to be better informed about their work environment (Newman 1994)

Building a theory involves discovery of new knowledge in the field of study and can be seen as rarely contributing directly to practice. Two approaches; deductive and inductive reasoning can be used in the process of theory building. In the positivist or deductive approach a general theory is generated and narrowed down to more specific hypotheses for testing and for collection of observations to address hypotheses. This ultimately leads to the test of the hypotheses with specific data or confirmation of the original theories. By contrast in the inductive research theory development begins with specific observations and measures, to detect patterns and regularities, formulate some tentative hypotheses for exploration, and finally end up developing some general conclusions or theories. Testing can be conducted in more or less natural settings using interpretive and pseudo-scientific (positivist) approaches. An interpretive study provides a more flexible setting for applied exploration without any predefined variables, but focusing on the full complexity as the situation emerges (Kaplan and Maxwell, 1994,). On the other hand, experimentation requires a certain level of control over some of the variables under consideration at least to the extent that independent, dependent and controlled variables can be predefined during the research design stage.

5.2 Research Methods

5.2.1 Research Methods Available

As there are various philosophical assumptions, which can inform research approaches, so there are various research methods. A research method is a strategy of inquiry, which moves from the underlying philosophical assumptions to research design and data collection and analysis. The choice of a specific research method is independent of the underlying philosophical position adopted. For example, case study research can be positivist (Yin, 1994), interpretive (Walsham, 1993), or critical, just as action research can be positivist (Clark, 1972), interpretive (Elden and Chisholm, 1993) or critical (Carr and Kemmis, 1986)

Fellow and Lie 1997 has listed five research methods that can be applied to research in construction: action research, case study research, ethnography, grounded theory, experiments and survey. A literature of the five and other research methods are reviewed in the following section

5.2.1.1 Action Research

Action research is a practical problem solving orientated research method capable of expanding the scientific knowledge (Nunamaker *et al.*, 1991; Mathiassen, 1997, Vidgen *et al.*, 1997; Baskerville & Pries-Heje, 1999). According to Argyris (1985), action research is a process of critical inquiry that fosters the learning of theory and practice. Although there are many different forms of action research (Baskerville, 1999; Baskerville and Wood-Harper, 1998), all are based on collaboration between researchers and practitioners.

Actions research is described as a five phase cyclic process (Susman & Evered, 1978), which involves, first the establishment of research environment that is a specification of for example the boundaries, scope and condition of the research followed by five iterative cyclic activities, including problem diagnosis, action intervention, evaluation and reflective learning. These are briefly described in the following section.

(1) Diagnosing. corresponds to the identification of the primary problems that are the underlying causes of the desire for change. Diagnosing involves self-interpretation of the problem, not through reduction and simplification, but rather in a holistic fashion. This

results in a development of theoretical framework (assumptions i.e., a working hypothesis) about the nature of the problem in a domain. (2) Action planning: This activity specifies actions that should improve or solve these primary problems. The theoretical framework, which indicates both, the desired future state and the changes that would achieve such a state guide the discovery of the planned actions. (3) Action taking then implements the planned action causing certain changes to be made. (4) Evaluation includes determining whether the theoretical effects of the action were realized, and whether these effects brought solutions to the problems. (5) In the learning cycle, the knowledge gained in the action research (whether the action was successful or unsuccessful) can be directed to wards diagnosis for further action research interventions or providing future directions for researchers.

5.2.1.2 Case Study

The term "case study" has multiple meanings. It can be used to describe a unit of analysis (e g. a case study of a particular organisation or situation) or to describe a research method Case study as a research method, Yin (1994) defines it as an empirical inquiry that: investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident" (Yin 1994,), and in which multiple sources are used. Case study research can be positivist, interpretive, or critical, depending upon the underlying philosophical assumptions of the researcher. Yin (1994) and Benbasat et al. (1987) advocates of positivist case study research, whereas Walsham (1993) advocate of interpretive indepth case study research, or critical, where research is mainly concerned with social critique (Habermas, 1984). Case studies are generalisable to theoretical propositions, the investigators goal is to generalise theories and not to generalise statistics. Data translated ideographically in terms of particular case, rather than nomothetically in terms of law like generalisation (Simister, 1995). In literature the use of case studies is described (Bloomfield 1998) in construction IT and (Simister 1995) in construction management studies. Case studies are often referred to interchangeably with ethnography, field study, and participant observation. Case studies require a problem that seeks a holistic understanding of the event or situation in question using inductive logical reasoning from specific to more general terms. Case study research method is particularly well suited to IS or IT related researches, since the object of IS discipline is the study of information systems in organizations, and "interest has shifted to organizational rather than technical issues" alone (Benbasat et al. 1987).

5.2.1.3 Ethnographic Research

Ethnographic research comes from the discipline of social and cultural anthropology where the researcher/ethnographer spends a significant amount of time in the field. The fieldwork notes and the experience of living there become an important addition to any other data gathering techniques that may be used. Ethnographers immerse themselves in the group they study (Lewis 1985,) and seek to place the phenomena studied in their social and cultural context. After early ground-breaking work by Wynn (1979), Suchman (1987) and Zuboff (1988), ethnography has now become more widely used in the study of information systems in organizations, from the study of the development of information systems (Hughes et. al, 1992; Orlikowski, 1991; Preston, 1991) to the study of aspects of information technology management (Davies, 1991; Davies and Nielsen, 1992) Ethnography has also been discussed as a method whereby multiple perspectives can be incorporated in systems design (Holzblatt and Beyer, 1993) and as a general approach to the wide range of possible studies relating to the investigation of information systems (Pettigrew, 1985).

5.2.1.4 Grounded Theory

Grounded Theory or as it was originally titled 'The Discovery of Grounded Theory' (Glaser and Strauss, 1967) is a method for the collection and analysis of qualitative data. Grounded theory is "an inductive, theory discovery methodology that allows the researcher to develop a theoretical account of the general features of a topic while simultaneously grounding the account in empirical observations or data" (Martin and Turner 1986). The major difference between grounded theory and other methods is its specific approach to theory development; grounded theory suggests that there should be a continuous interplay between data collection and analysis. Grounded theory approach is extremely useful in developing context-based, process oriented descriptions and explanations of the phenomenon (Orlikowski, 1993).

In this method conceptual properties and categories may be discovered or generated from the qualitative data by following a number of guidelines and procedures. The three concepts of Grounded Theory have resonance with the process of interpretive research.

Firstly *constant comparative analysis*: a procedure for identifying conceptual categories and their properties that may be embedded in the data.

Secondly *theoretical sampling:* by which the conceptual categories are enriched through coding and integration.

***Thirdly*:** these two procedures above lead to the development of a hierarchy of integrated categories, and to the emerging theory. *Theory* is the third concept of note, and the usage of the term is critical. Strauss and Corbin (1994) maintain that theory consists of

> "Plausible relationships proposed among concepts and sets of concepts researchers are interested in patterns of action and interaction between and among various types of social units (i.e. actors) They are also much concerned with discovering process not necessarily in the sense of stages or phases, but in reciprocal changes in patterns of action/interaction and in relationship with changes of conditions either internal or external to the process itself" (Strauss and Corbin, 1994)

and they note two important features: Firstly that they are traceable to the data and secondly that they are 'fluid', that is to say the emphasis is on process and the temporal nature of the theory. So then 'theory' is used in the method to refer to local empirical models surrounding the phenomenon under study, it is not substantive. The theory is made apparent through the production of an account and/or associated relationship diagrams of categories. Dey (1999) provides useful insight into the term:

> "Theory focuses on how individuals interact in relation to the phenomenon under study; it asserts a plausible relation between concepts and sets of concepts; it is derived from data acquired through fieldwork interviews, observations and documents; the resulting theory can be reported in a narrative framework or as a set of propositions" (from Dey, 1999,).

5.2.1.5 Concept Mapping

Conceptual mapping technique began during the sixties with the Joseph's D. Novak studies (1993) at the Cornell University. His job was based on David's Ausubel theories (1968) that emphasized the importance of the ability to learn new concepts. Conceptual mapping was born therefore for being able to formalize structured knowledge: in other words the way in which several concepts are correlated inside a determined cognitive dominion.

Concept mapping is a type of structured conceptualisation, which can be used by groups or individuals to develop a conceptual framework that can guide evaluation, planning etc. In a typical case, six steps are involved:

Preparation Step: involves identification of participants and development of focus for the conceptualisation.

Generation Step: development of statements that address the focus

Structuring Step: sorting of the statements into piles of similar ideas or structuring of statements

Representation Step: the representation of statements in the form of a concept map. This is the process of taking the sort and rating input and "representing" it in a map form using multidimensional scaling and cluster analysis. There are two major statistical analyses that are used in this process. (1) The first, multidimensional scaling takes the sort data across all participants and develops the basic map where each statement is a point on the map and statements that were piled together by more people are closer to each other on the map. (2) The second analysis cluster analysis takes the output of the multidimensional scaling (the point map) and partitions the map into groups of statements or ideas, into clusters. If the statements describe activities of a program, the clusters show how these can be grouped into logical groups of activities. If the statements are specific outcomes, the clusters might be viewed as outcome constructs or concepts.

Interpretation Step: this is the stage where individual labels and interpretations of concepts are developed for various maps.

Utilization Step: involves using the maps to help address the original focus. On the program side, the maps can be used as a visual framework for operational program. On the outcome side, they can be used as the basis for developing measures and displaying results

Trochim and Linton (1986) proposed a general framework for structured conceptualisation and showed how specific conceptualisation processes can be devised to assist groups in the theory and concept formation stages of planning and evaluation. Conceptualisation refers to the articulation of thoughts, ideas, or hunches and the representation of these in some objective form. While in a planning process, the aim is to conceptualise the major goals and objectives, needs, resources and capabilities or other dimensions, which eventually constitute the elements of a plan In evaluation, the aim is to conceptualise the programs or treatments, samples, settings, measures and outcomes, which are relevant.

Concept mapping is used for many purposes strategic planning, product development, market analysis, decision-making, measurement development, research formulation etc. In concept mapping, ideas are represented in the form of a picture or map. To construct the map, ideas first have to be described or generated, and the interrelationships between them articulated. Multivariate statistical techniques, multidimensional scaling, and cluster analysis are then applied to this information and the results are depicted in map form. The method generates a group aggregate map; it utilizes multivariate data analyses to construct the maps, and it generates interval-level maps, which have some advantages for planning and evaluation, especially through pattern matching.

In concept mapping, it is important to distinguish between theories and concepts. One thing to recognize is that while theories are built upon concepts, concepts are not, in and of themselves, theories A theory postulates a relationship usually causal between two or more concepts. A concept map provides a framework within which a theory might be stated Thus, concept maps can act as the framework for a statement of theory, but are usually not considered a theory in and of themselves

Group concept mapping is consistent with the growing interest in the role of theory in planning and evaluation. In evaluation, for instance, this interest is evidenced in writings on the importance of program theory (Bickman, 1986; Chen and Rossi, 1983, 1987); in the increased emphasis on the importance of studying causal process (Mark, 1986); in the recognition of the central role of judgment especially theory-based judgment in research (Cordray, 1986; Einhorn and Hogarth, 1986), and, in the thinking of critical multiplism (Shadish et al, 1986) which emphasizes the role of theory in selecting and guiding the analysis of multiple operationalizations. Concept mapping can be viewed as one way to articulate theory in these contexts. In planning, conceptualisation has had somewhat more attention and is evidenced in the sometimes daunting proliferation of different planning models and methods of conceptualisation (Dunn, 1981).

5.3 Modes of Analysis

5.3.1 Approaches to Data Analysis

In this section three approaches to data analysis for qualitative research, are reviewed, these are: the hermeneutics, semiotics, and approaches, which focus on narrative and metaphor. Grounded theory is also a mode of analysis and research method (Martin and Turner 1986). These modes of analysis are different approaches to gathering, analysing and interpreting qualitative data. The common thread is that all qualitative modes of analysis are concerned primarily with textual analysis (whether verbal or written). Hermeneutics and semiotics can be treated as both an underlying philosophy and a specific mode of analysis.

Hermeneutics

In an organization, people (e.g. different stakeholders) can have confused, incomplete, cloudy and contradictory views on many issues. The aim of the hermeneutic analysis becomes one of trying to make sense of the whole, and the relationship between people, the organization, and information technology. Hermeneutics can be treated as both an underlying philosophy and a specific mode of analysis (Bleicher, 1980) As a philosophical approach to human understanding, it provides the philosophical grounding for interpretivism. Hermeneutics, as a mode of analysis suggests a way of understanding textual data It is primarily concerned with the *meaning* of a text or text-analogue (an example of a text-analogue is an organization, which the researcher comes to understand through oral or written text) The basic question in hermeneutics is what is the meaning of this text (Radnitzky 1970). Taylor says that:

> "Interpretation, in the sense relevant to hermeneutics, is an attempt to make clear, to make sense of an object of study. This object must, therefore, be a text, or a text-analogue, which in some way is confused, incomplete, cloudy, and seemingly contradictory in one way or another, unclear The interpretation aims to bring to light an underlying coherence or sense" (Taylor 1976).

The idea of a hermeneutic circle refers to the dialectic between the understandings of the text as a whole and the interpretation of its parts, in which descriptions are guided by anticipated explanations (Gadamer 1976). As Gadamer explained, " The movement of understanding is a circular relationship, the anticipation of meaning in which the whole is envisaged becomes explicit understanding in that the parts, that are determined by the whole, themselves also determine this whole." Ricoeur suggests, "Interpretation, is the work of thought which consists in deciphering the hidden meaning in the apparent meaning, in unfolding the levels of meaning implied in the literal meaning" (Ricoeur 1974).

Semiotics

Semiotics is primarily concerned with the meaning of signs and symbols in language. The essential idea is that words/signs can be assigned to primary conceptual categories, and these categories represent important aspects of the theory to be tested. One form of semiotics is content analysis Krippendorf (1980) defines content analysis as a research technique for making replicable and valid references from data to their contexts. The researcher searches for structures and patterned regularities in the text and makes inferences on the basis of these regularities. Another form of semiotics is conversation analysis In conversation analysis, it is assumed that the meanings are shaped in the context of the exchange (Wynn, 1979). The researcher immerses himself/herself in the situation to reveal the background of practices. A third form of semiotics is discourse analysis Discourse analysis builds on both content analysis and conversation analysis but focuses on language games. A language game refers to a well-defined unit of interaction consisting of a sequence of verbal moves in which turns of phrases, the use of metaphor and allegory all play an important part.

Narrative and Metaphor

There are many kinds of narrative, from oral narrative through to historical narrative. Metaphor is the application of a name or descriptive term or phrase to an object or action to which it is not literally applicable (e.g a window in Windows 95). Narrative and metaphor have long been key terms in literary discussion and analysis. In recent years there has been increasing recognition of the role they play in all types of thinking and social practice. Scholars in many disciplines have looked at areas such as metaphor and symbolism in indigenous cultures, oral narrative, narrative and metaphor in organizations, metaphor and medicine, metaphor and psychiatry etc

5.4 Valid Analysis

5.4.1. Criteria for Evaluating Research

According to Johnson (1997), validity was traditionally attached to quantitative research; but what it means when qualitative researcher speaks of validity is qualitative research that is plausible, credible, trustworthy and defensive. Most validation methods were developed for quantitative research, but have also been applied to qualitative research. Some qualitative researchers refer the issues of validity as establishing adequacy of evidence and credibility (Chenitz & Swanson, 1986). Guba and Lincoln (Trochim 2001) proposed four criteria for evaluating the soundness of qualitative research, an alternative to the more quantitatively oriented criteria: credibility or authenticity (Trochim, 22001) for quantitative criteria of internal validity; transferability, (Trochim, 2001 & Miles & Huberman, 1994) for quantitative criteria of external validity; dependability or auditability (Miles & Huberman, 1994 & Trochim, 2001) for quantitative criteria of reliability; and finally, conformability is a qualitatively oriented criterion for objectivity, this criterion refers to the degree to which the results could be confirmed or corroborated by others (Trochim,2001).

Burstein & Gregor proposed five criteria by which software development (SD) type research might be evaluated (Burstein F. & Gregor S. 1999). It is argued that SD research is a form of action research and will often employ a case study approach and qualitative data collection techniques may be used. The criteria are drawn mainly from discussion of case studies, action research and qualitative studies (Miles and Huberman 1994; Yin 1994). Although these criteria are proposed for this type of research work, it is not argued that these criteria are, at an underlying level, markedly different from criteria used for more traditional work in experimental or quasi-experimental settings (Cook and Campbell 1979) the underlying concerns for the validity of claims and arguments made should be similar. Rather, the criteria may use different terms or have different emphasis. These criteria are: Significance, Internal validity, External validity, Objectivity/Conformability, and Reliability/Dependability/Auditability, which are reviewed as follows:

1) Significance.

The study must be significant, either theoretically, practically, or both. Yin (1994) in discussing case studies says that "the individual case or cases are unusual and of general public interest. The underlying issues are nationally important, either in theoretical terms or in policy or practical terms or they are both the preceding".

Some relevant queries that might be asked are:

- Does the study have theoretical significance? Is it theory-building or theory-testing?
- Does the study have practical significance? Will it contribute to the building of "better" solution?

2) Internal validity

Internal validity refers to the credibility of the arguments made. Do the findings of the study make sense? In experimental and quasi-experimental work, internal validity refers to the nature of the evidence and arguments for causal relationships between constructs (Cook and Campbell, 1979). With respect to interpretive work, the importance of such aspects as "apparency" and "verisimilitude", "authenticity", "plausibility," and "adequacy" are stressed (Miles and Huberman 1994). Yin (1994) suggests that in collecting data about a system in a case study one needs to ask if the study is complete. The boundary of the study should be defined in such a way that it can be seen that information beyond the boundary is of decreasing relevance. In addition, the reader should be convinced that very little relevant evidence remains untouched.

3) External validity

Cook and Campbell (1979) define external validity as approximate validity with which conclusions are drawn about the generalizability of a causal relationship to and across populations or persons, settings, and times Some relevant queries that might be asked of an SD study are:

- Are the findings congruent with, connected to, or confirmatory of prior theory?
- Are the methods, processes and outcomes described in conclusions generic enough to be applicable in other settings?

- Does the researcher define the scope and boundaries of reasonable generalization from the study?
- Is the transferable theory from the study made explicit?

4) Objectivity/Conformability

Miles and Huberman (1994) say that an important basic issue is one "of relative neutrality and reasonable freedom from unacknowledged researcher biases at the minimum, explicitness about the inevitable biases that do exist". Other authors may class this criterion under the heading of internal or construct validity (Cook and Campbell 1979). Some relevant queries that might be asked of an SD study are:

- Are the study's methods and procedures described explicitly and in detail?
- Can we follow the procedures of how data was collected?
- Has the researcher been explicit and as self-aware as possible about personal assumptions, values and biases? This question is of special importance in action research. For example, is there a likelihood of bias if people testing a system or method are aware of the identity of the developer? Could feelings about the developer influence responses to questions about the system?

5) Reliability/Dependability/Auditability

Miles and Huberman (1994) state that the underlying issue here is whether the process of the study is consistent, reasonably stable over time and across researchers and methods. The question is one of quality control. Have things been done with reasonable care. Relevant queries include:

- Are the research questions clear?
- Is the researcher's role and status explicitly described (of particular interest in action research)?
- Are basic constructs clearly specified?

5.5 Research Strategy and methods of the Study

5.5.1 Theoretical Framework

The interpretive position of underlying epistemology, (Boland 1985, Walsham 1993) which believe that knowledge of reality is gained through social construction such as language, consciousness, shared meaning, guides the research methodologies and approaches used for this research study. Context-dependent knowledge has been colleted and analysed in an iterative manner until context independent representations of the knowledge are achieved through compromise in the conceptual mapping of the shared knowledge across the contexts, to a common representations/definitions provided by a neutral language, the Process Specification Language (PSL).

Prior to the adoption of PSL in the construction environment it has become important to investigate the applicability of the language in the construction environment and evaluate the effectiveness and usefulness of the language in overcoming the information exchange problems that prevail between construction software applications, provided that that language is proved applicable. Therefore, this research study is carried out in three sets of major phases, these are: literature review; investigation of the applicability of the Process Specification Language (PSL), which is developed for manufacturing software interoperability, in construction environment; and finally implementation of the language in a constriction scenario.

In the **first phase** of the study, an intensive literature review was carried out in the area of developments and researches conducted by standardisation, industrial and academic bodies for software interoperability, information standardisation; information and processes integration; and the research methodologies used in the construction and manufacturing environments.

Phase two involved the investigation of the applicability of the PSL language in the construction environment through conceptual mapping of the construction process concepts within the pre-construction process stages to PSL ontology, which contains neutral specification/definitions for manufacturing process concepts. In this phase an in depth literature study of PSL language and technical analysis of concepts and constructs were carried out to identify the manufacturing process concept defined in the PSL ontology and deduce similarities and differences between the manufacturing and construction processes. This is followed by a technical analysis on a set of three construction software applications, supporting the pre-construction process stages, which was chosen as a test scenario for the applicability of PSL in construction. During the technical analysis of the software applications a set of data types that are necessary for each of the software applications to represent construction process information were identified and gathered. Based on the data types gathered, conceptual models that are centred around the basic data entities/types were developed for each of the software applications and the attributes associated with each basic data entity were identified and categorised. Finally, conceptual mapping of the data types gathered to the PSL ontology were conducted through multidimensional scaling/conceptual analysis between the applications data types and the concepts defined in the PSL ontology **Figure 5.1** shows the mapping of the applications data types to the PSL ontology parts or to the new extensions. This is to identify and determine the parts of the PSL ontology that are applicable to providing the required common definition for the shared sets of the contextual data types or construction process concepts that are shared among the three software applications.

PSL ontology is a map of process concepts, which is developed, in the manufacturing environment thorough conceptual mapping of process concepts identified and gathered from different manufacturing software applications and application areas. By conceptual mapping of the construction process concepts to PSL ontology, the intention of the researcher, is not to propose a modification or alteration to the PSL ontology it is rather, based on the applicability of the PSL language, to use the PSL definitions to the representations of the construction process concepts. In the concepts mapping process, of this research study, based on the similarity between the data types representing construction process concepts and PSL ontology concepts the multidimensional conceptual analysis creates mapping points, which map the set of data types that were identified and gathered during the technical analysis of the scenario software applications, to the PSL map (PSL ontology).

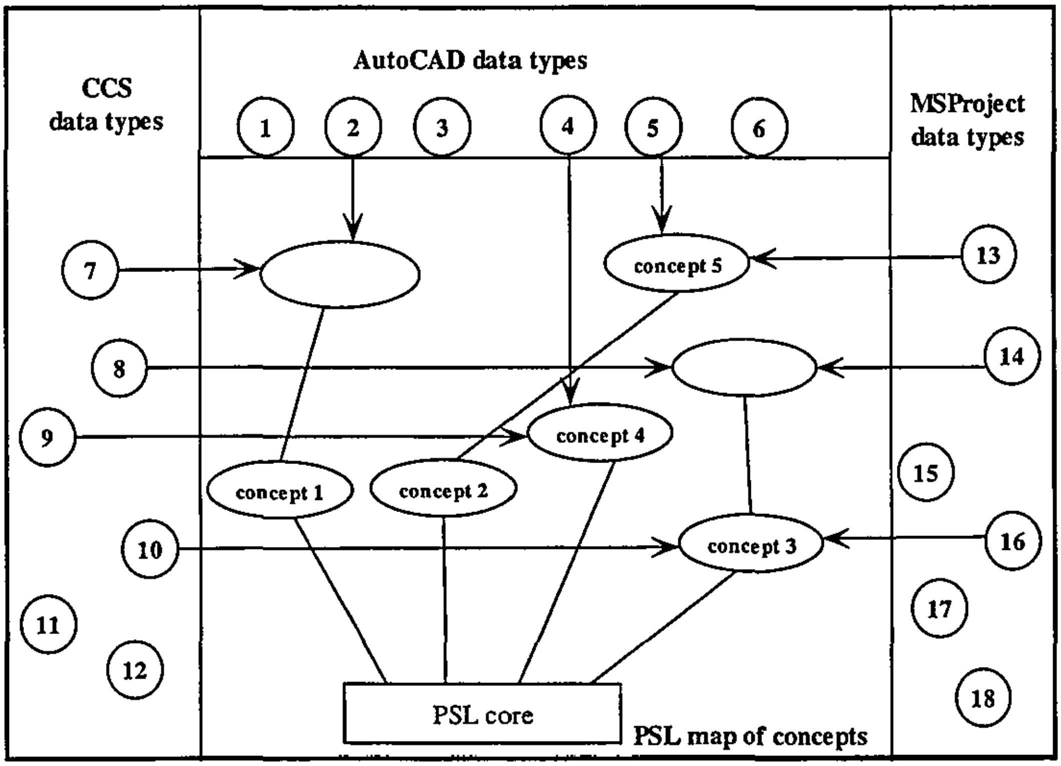


Fıgure 5.1. Conceptual mappıng of the software applicatıons' data types to PSL ontology

Figure 5.1 Depıcts the mappıng process of the software applications' data types to PSL ontology, ın which the taıl of the arrows ındıcate the software applıcation' data types, and the head of the arrows poınts to the PSL parts that contaın concepts that are applıcable to define the data types at the taıl of the arrow. Where the arrowhead point to an empty cırcle, thıs ındicates that there ıs no concepts defined in PSL ontology that is applıcable to the data type at the taıl of the arrow. Hence thıs empty cırcle ıs an open space for proposal of new extensıons to ıncorporate new constructıon process concepts in the PSL ontology that can provıde neutral definitıons for the data types at the arrow taıl. The actual mappıng process is reflected ın **chapter 7.**

Based on results from the investıgation of the applıcabılity of PSL in the constructıon envıronment, in the **third phase** of the research study, the evaluation of the effectıveness and usefulness of the language in the construction environment was conducted through implementation of the language in a construction scenario. The construction scenario is described in **chapter 7** and the description of the construction case study project for the scenario and the implementation of the PSL is in the scenario is presented in chapter 8. In this phase, a rigorous and systematic intervention process is required to solve the practical problems of this research study. One acceptable way to systematically evaluate the effectiveness or usefulness (or the effective use) of the PSL language in construction is through implementation of the language in a construction scenario. The problem of evaluating the effectiveness or usefulness of PSL in a construction scenario is of theoretical and practical importance The implementation of the PSL language in a construction scenario provides the framework for the intervention into construction exchange scenario and it guided the investigation and critical analysis of the research problem. The interest of the research is in improving IT and communication practices so the motivation was to solve the problems in a manner that contributed to the existing body of knowledge about research in construction IT. To achieve this objective of the study parsing and concept mapping or pattern matching were used as the methods of choice in the implementation of the PSL language in the construction scenario, because an interventionist research method that would advance knowledge in communication practice is needed. It is identified that the implementation of the language in a real type construction case study is applicable to achieve the objective of this study due to its practical problem solving orientation and its ability to expand the knowledge about construction IT problems and possible solutions.

The implementation of the language in the construction scenario was conducted through translation of the scenario software applications' representations of the case study project construction process information to PSL representations and back to the applications. This translation involves a two stages process: syntactic translation and semantic translation. The syntactic translator is a parser between the PSL syntax (e.g. KIF) and the native syntax of one of the applications; this parser keeps the terminology of the

Translation Processes

While in the syntactic translation process, the applications' process information representations of the case study project were translated to PSL syntactic representations, which is a Knowledge Interchange Format, (KIF) syntax using a parsing method. The parsing method, translates each of the application's syntax to PSL syntax keeping the applications terminology in tact. In the semantics translation process, the case study project data in PSL syntax, which is result of the parsing, was translated to PSL semantic representations through mapping or pattern matching of the case study project data in PSL syntax to PSL semantic concepts. The actual translation process is carried out by hand using parsing procedures and hand written semantic translators (semantic translation definitions written between the applications and PSL) and the result of these translations is elaborated in **chapter 8.**

Construction Scenario within Pre-construction Process Stages

A multidimensional conceptual analysis and conceptual mapping research were conducted in a construction scenario within the pre-construction process stages that involves process information representation and exchange using the candidate construction software applications supporting these stages. A construction scenario within the pre-construction process stages was chosen because it is in these stages of construction projects that most of the construction process related information is generated and manipulated and these stages are one of the primary area in which software applications are wieldy used to support the generation and manipulation of the construction process related information. Additionally, this area exhibited the complexity of the process related information that are manipulated and shared among the stages and the problem of exchanging this information between the software applications supporting these stages. Therefore it is considered as an appropriate test scenario for evaluating the effectiveness/usefulness and the impact of the PSL language in the Architecture Engineering and construction practices.

The construction scenario comprised three process stages and supporting software applications, a construction case study project, and case study project construction process information exchange between the scenario supporting software applications. The three stages involve, architectural design, pre-tender cost estimating, and scheduling processes for the construction case study project and the supporting candidate software applications are AutoCAD design, CCS cost estimating, and Microsoft project scheduling.

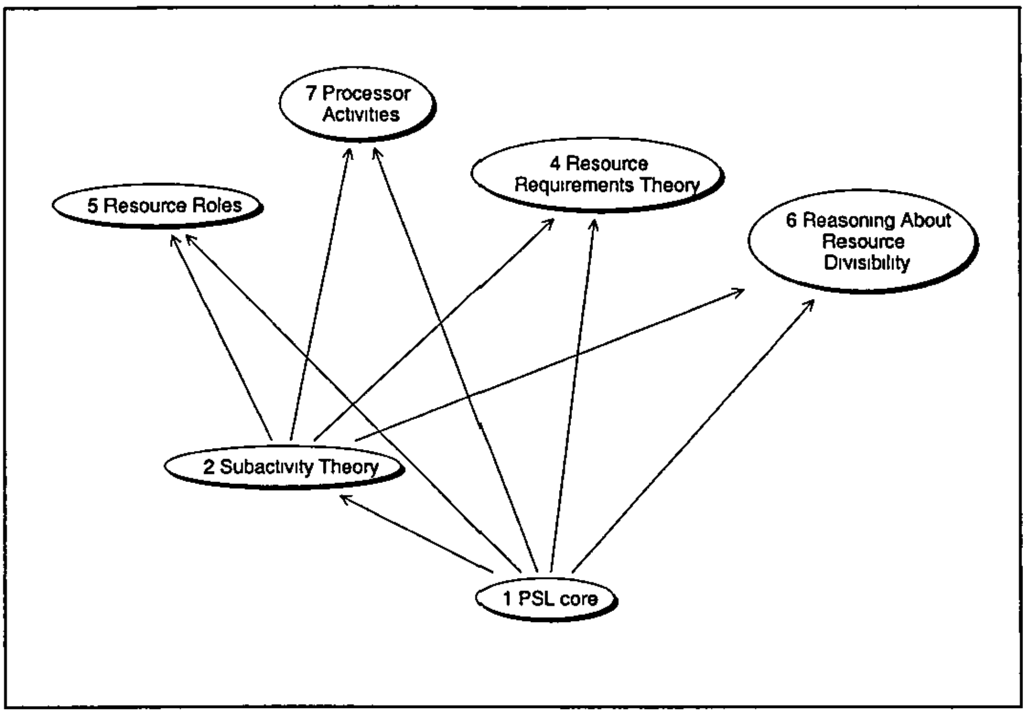

7 Processor Activities
4 Resource Requirements Theory
5 Resource Roles
6 Reasoning About Resource Divisibility
2 Subactivity Theory
1 PSL core

Figure 5.2. PSL ontology component parts of interest to this project

The Research Problem

The aim of the research study is to evaluate the effectiveness/usefulness of the PSL, an existing process specification and exchange language, based on its ability to interoperate manufacturing software applications, to solve the prevailing interoperability problems among process related construction software applications. PSL is ontology or a data model, which incorporates a syntax and grammar specification to make it a language. PSL ontology component parts of interest to this project are illustrated in **Figure 5.2.** The PSL Core is a set of axioms written in the basic language of PSL. PSL core is a relatively simple theory with purpose to axiomatize a set of intuitive semantic primitives that is adequate for describing a wide range of basic processes. A foundational theory is a theory whose expressive power is sufficient for giving precise definitions of, or axiomatizations for, the primitive concepts of PSL, thus greatly enhancing the precision of semantic translations between different schemes. The final component of PSL consists of PSL *extensions* that give the resources to express information involving concepts that are not part of PSL core. The relationship between the PSL components are shown by arrows, while the solid arrows indicate the definability relation, the dashed lines indicate partial definability, i.e., the case where some, but not all the additional linguistic items in the language of an extension are definable. **Figure 5.2** shows the PSL map of manufacturing process concepts and theories. To reduce congestion of concepts and theories only few of the PSL parts that are considered to be of interest to this research study are presented in the map.

The Information Exchange Problems in the Pre-construction Process Stages

The aim of this research study is to improve the construction processes by solving the organizational and technical problems: that is the lack of shared semantics or understanding among the process related construction software applications It was believed that by using PSL as a process specification and exchange language in construction scenarios, the construction software applications would have better understanding of each other and exchange the correct information. Prior to the implementation of the PSL language in this study there was no semantic understating between the software applications supporting the scenario within pre-construction process stages. This is because each of these work in a stand-alone manner with unique representations and terminologies, therefore, do not share the same process concepts, even when the same terminology is used, or different terminologies may be used for the same process concept Consequently, a decision was made to implement the PSL language in the scenario within the pre-construction process stages and it is expected to prove the usefulness or positive impact of the language in the construction environment.

The Effectiveness of PSL for Construction Process Specification and Exchange

The effectiveness/usefulness of the language in construction is measured by its ability to solve the prevailing problems of information exchange hence by its applicability to facilitate the correct and complete exchange of process related information between construction software applications in the environment. The main goal is to find out how useful and effective is the PSL language for construction process specification and exchange thorough implementation of the language in construction scenario. The result of the implementation exercise is expected to provide, significant support for evaluating how effective and useful the language for construction organisations, and basis for future directions in the use of PSL in the sector.

The implementation of the PSL in the construction scenario required identification and formal representation of the construction process related data of the case study project. This is achieved, after identification and gathering of process related data from the construction specification document for the case study project, first through process models development for the case study scenario project construction, using IDEF3 modelling methodology. This modelling facilitated the formal representation, of the processes required to construct the case study project and the process data related to the three software applications supporting the scenario. Then the three software applications were used to represent the processes and process related data of the project. Finally, using PSL as an interchange language, the two stages translation process is carried out to exchange process information between the three scenario software applications. The role of the two stages translation process in the implementation of PSL in construction case study scenario was to facilitate the step-by-step transfer of information from applications representations and semantics to PSL and back to the applications. The exchange of information using PSL involved translation of information between the software applications using hand written translators (translation definitions between the applications and PSL). Though, this translation process and concepts mapping has limitations in providing a significant support for evaluation of how effective and useful the language is in construction, it supported the identification of the applicable PSL concepts in the scenario and formulation of extensions to the PSL ontology that are needed to incorporate concepts that are specific to the construction scenario through identification of the concepts that were not supported by PSL and their relations to other parts of the PSL ontology that were used in the translation process. Finally the result of the implementation exercise was evaluated through analysis of the scenario before and after the implementation of PSL, to measure the success of the implementation and the impact of PSL in construction information exchange within the scenario and the proposal for new extensions to the language was formulated. **Figure 5.3** Shows the construction scenario before and after the implementation of PSL and strategies in the implementation of PSL in the scenario.

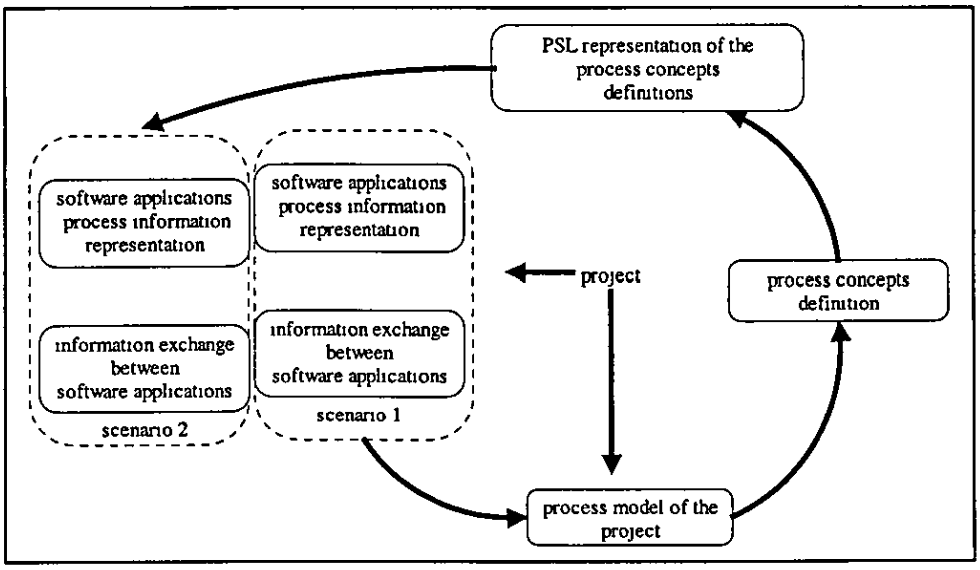

PSL representation of the process concepts definitions
software applications process information representation
information exchange between software applications
scenario 2
software applications process information representation
information exchange between software applications
scenario 1
project
process concepts definition
process model of the project

Figure 5.3. Scenarios 1 before and 2 after the implementation of PSL

5.6 Research Methods Used

A technical analysis, multidimensional conceptual analysis, conceptual mapping, process modelling, translation processes and a single case study, have been used in this study to investigate different aspects of the research problem as described below.

1. A technical analysis of the construction scenario software applications was used to identify the data representations: data representation terminologies adopted by each application, file formats and common exchange file formats through which the applications may exchange physical files, to identify and gather the data types that are necessary for representation of construction processes information, and to identity and determine a common set of data types representing the same process concepts or information that are shared among the applications.
2. Modelling of the processes for the case study project construction is used to formally represent the construction processes and process data, related to the scenario candidate software applications.
3. A conceptual mapping is used in the investigation of the applicability of PSL in construction. It is used to map the data types, gathered through technical analysis of the scenario software applications, to the PSL ontology based on the result of the

multidimensional conceptual analysis between the application's data types and PSL ontology. Using this method context dependent data types were mapped to PSL to find context independent representation (integrated data or neutral specification) for the common set of data types that are shared among the software applications.

4. A construction scenario within the pre-construction process stages, which involve process information representation and exchange between AutoCAD, CCS and Microsoft project applications for a case study project is used to explore the practices in construction; information representation and exchange, the methods of information exchange between participants, and the problems. The research is carried out within pure academic research with the candidate software applications supporting the construction process stages of the scenario were used to represent the various professional participants involved in the construction scenario to give a real construction project character of the industry. This includes the information generated, handled and exchanged between the participants' activities using IT applications to represent a real construction phenomena.
5. A manual translation processes is used in the implementation of PSL in the construction scenario. In this implementation a syntactic and semantic translation were carried out through parsing and pattern-matching and mapping methods. This includes development of hand written translators (translation definitions) between the software application and PSL and information exchange between the software applications using the hand written translators and finally validation of the implementation.
6. Additionally, an extensive literature review in the areas of research and developments and researchers for software interoperability was used as a secondary source of information for the research study.

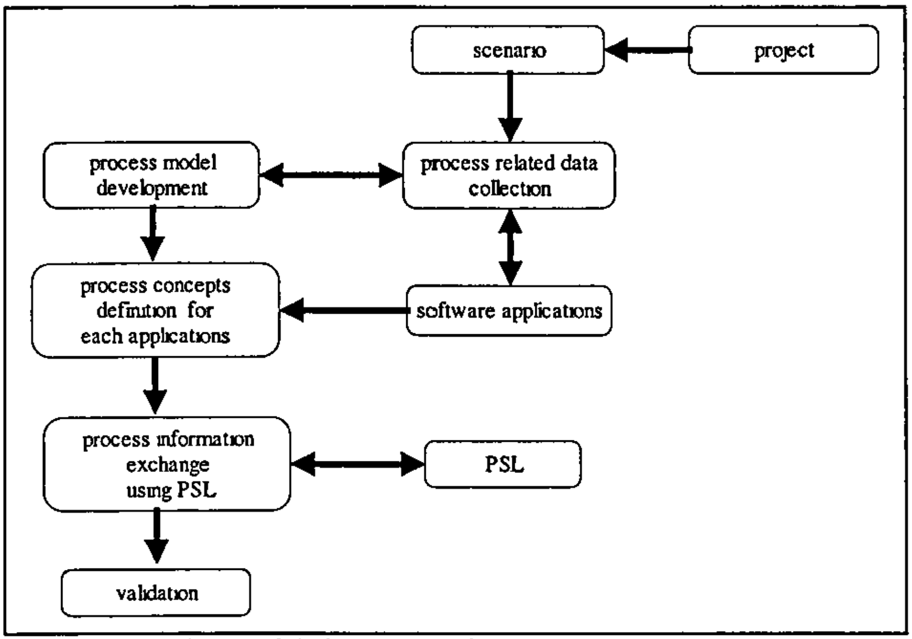

scenario
project
process model development
process related data collection
process concepts definition for each applications
software applications
process information exchange using PSL
PSL
validation

Figure 5 4. Summary of the research strategy

5.7 Data Collection and Analysis Methods Used

The research problems have been addressed through combination of qualitative research methods and data gathering and analysis methods. Although a clear distinction between data gathering and data analysis is commonly made in quantitative research, such a distinction is problematic for many qualitative research (Bleicher, J., 1980). For example, from a hermeneutic perspective it is assumed that the researcher's presuppositions affect the gathering of the data. The analysis affects the data and the data affect the analysis in significant ways. In qualitative research, it is therefore more appropriate to talk of "modes of analysis" rather than "data analysis" (Bleicher, J. 1980). These modes of analysis are different approaches to gathering, analysing and interpreting qualitative data. The common thread is that all qualitative modes of analysis are concerned primarily with textual analysis (whether verbal or written).

Mode of Analysis

Different modes of analysis were used in the different part of the research to address the research problems Technical and empirical data have been collected with respect to;

technical characteristics of the construction scenario candidate software applications, data representation terminologies and data types of these applications, process concepts in each component parts of the PSL ontology, construction case study project data, construction process information of the case study project related to scenario within the pre-construction process stages. The methods used in the data collection and the mode of analysis and the purposes are summarised in **table 5.5.**

Components	**Technical Study/Research**	**Qualitative Research**
Purpose	To identify the common or exchange file formats, through which the scenario software applications may exchange physical files; To identify and gather process related data types and terminology of the software applications; To identify and determine the data types of the applications repressing the same concepts; To identify and determine PSL concepts and theories applicable in construction, and investigate the level of applicability of PSL concepts and theories for defining the shared process related data types between the three software applications	To identify the processes and process related data from the case study project for a formal representation of the construction process information using IDEF3 modelling method, To identify contextual process data of the case study project i e data related to each of the scenario applications and define the process concepts using the software applications representations or data types, and finally, to define these concepts using PSL language through mapping of these concepts to PSL ontology.
Data collection method	Technical	Qualitative data collection method
Source of data	Scenario Candidate Software applications and technical documents; and PSL ontology	Case study project and scenario applications' data types
Mode of analysis	Technical analysis	Qualitative analysis; Semiotics,

Table 5.2. Summaries of the data collection and analysis methods and the purposes

In the assessment of the software interoperability problems, sets of technical data. information representation data types and terminologies, file formats, and import/export or file exchange file formats, were gathered from the construction scenario software applications and a technical analysis were conducted to identity the common file formats that these applications may use to interoperate, and the data types and terminologies representing the shared concepts between the applications.

In the investigation of the applicability of PSL in the construction environment, conceptual properties and categories were generated from the technical data gathered through technical analysis of the scenario software applications and PSL ontology, and a multidimensional conceptual and technical analysis were conducted to identity and determine the parts of PSL ontology that are applicable in providing common definitions for the shared concepts between the applications, and the new concepts that may need to be incorporated in PSL ontology new extensions.

In the formal representation of the case study project construction process information, using IDEF3 modeling methodology, data were gathered from the construction specification document of the case study project, in correlation with the scenario software applications and qualitative and technical analysis were conducted to identity the construction processes and corresponding data related to the three scenario software applications.

Understanding of the construction scenario situations involved, gathering and analysis of the whole data related to the construction processes of the case study project; identification and collection of contextual process data related to each of the scenario software applications, identification and collection of data types of the software applications representing the case study project construction process information, the problems in exchanging this information between the scenario software applications required identification of the file formats of each of the applications and the common or exchange file formats for the exchange of physical files between the applications, and the level of semantic understanding and/or misunderstanding that may exist between the applications representations or data types.

In the analysis of the data gathered during the analysis of the construction scenario process concepts and manufacturing process concepts of the PSL ontology a multidimensional conceptual analysis were employed for referencing data in their contexts, shaping the meanings of each of the applications data types in their context and based on this units of interactions were carried out, in which is a sequence of data moves from context dependent (the applications data types) to context independent representation (PSL representations)of the data. This is to identify the semantic similarities and differences between the scenario software applications' process information representations (data types) and the PSL concepts and theories that are applicable for defining the data types, which represent shared process concepts between the three software applications.

5.8 Conclusions

The overall research problem was addressed through a combination of different approaches and methodologies studied and used in the different parts of the research. The situations in the complex contexts of IT applications in construction were analysed in terms of the level of applicability of the PSL language in the construction environment. This is achieved through investigation of the applicability of PSL in the construction environment and implementation of the language in a construction scenario. Finally the impact of the language in the construction scenario was evaluated through technical and conceptual analysis of the results of the implementation exercise

Chapter 6 Investigation of Applicability of PSL in Construction

6.1 Introduction

There are numerous computer applications within the construction industry, which deal with the description and representation of construction processes (management of construction process information) prior to the implementation of these processes on site Integrated computer systems offer the capability of improving the effectiveness and efficiency of construction management processes (Russell & Froese 1995) A central requirement for such systems is the ability to share information between multiple computer applications Therefore, the integration of these applications is cornerstone for an effective and efficient management of the construction operations. However, the ability to share process information between these applications is highly dependent on the specification and representation of the information. An integrated process related construction computer applications environment in the industry requires a common specification language to facilitate the representation and exchange of process information between heterogeneous software applications.

Ontology based processes description and interchange mechanisms have simplified the integration of some process related software applications in the manufacturing environment, one such mechanism is the Process Specification Language (PSL) (http://www mel.nist.gov/psl/pubs.html). Motivated by a growing need to share process information in the manufacturing environment, PSL was developed at National Institute of standards and technology (NIST), aimed at creating a generic Process Specification Language, focused on the description of processes building on existing modeling and process representation methods, for integration of multiple process-related applications throughout the manufacturing life cycle, (Schlenoff, C., Gruninger M., Tissot, F., Valois, J., Lubell, J., Lee, J., 2000). In the same way, there is an expectation that the construction sector could benefit from PSL for interoperability of multiple process-related software applications. One of the main objectives of this research study focuses on the assessment and investigation of the applicability of the Process Specification Language (PSL) in the construction environment.

This chapter reviews the process concepts defined in the PSL ontology and the construction environment from the production/construction processes perspective, looks briefly at the design, cost estimating and scheduling construction application areas and supporting software applications and discusses the process concepts (construction process concepts) of these application areas based on results of analysis of the software applications supporting these areas. Finally, the applicability of the PSL language in construction is assessed through a multidimensional conceptual analysis between the construction process concepts and PSL ontology (manufacturing process concepts), and mapping of the software applications data types, which deal with the representation and manipulation of construction process information within the pre-construction process stages, to PSL ontology

6.2 Construction Environment

6.2.1 Introduction

Construction is a project based industrial development where each project includes diverse work pieces and tasks, within each stage of projects lifecycle. The construction industry has not been at the leading edge of the industrial development except for one aspect and that is the ability to organise and re-reorganise it self around big projects (A. Jagbeck, 1998). In-order to overcome the unlimited and complex management needs the industry had to rely on production units, sub-contractors or work groups. Each work group is independent, and yet interdependent in the process of achieving a final goal, the constructed facility. As a result the need for communication and information exchange between the work groups is of paramount importance in the construction industry

Construction encompasses intensive and fragmented processes with numerous participants, experts and professionals involved across the various stages throughout the projects life cycle and large amount of information is generated for each project undertaken, and this information is also fragmented. The various participants usually come from several diverse organisations and involved in different phases throughout the project lifecycle with different goals, needs and cultures. These participants have their own unique terminology, technology applications and way of expressing and handling information and yet interdependent. Moreover, much of the information exchange between participants is still done manually or using information technology only in the rudimentary way.

6.2.2 Construction Process Information

Construction involves designing and implementation of a building project form the conception of the project in a client's mind to its completion for commissioning and use (CIB W65, 1985). For an effective and efficient management of construction projects the construction processes need to be described, represented and managed prior to the actual implementation of these processes on site Process descriptions include information describing construction processes such as in design, cost estimating and scheduling stages, and this is referred to as construction process information. The term "construction process information" refers to information describing the construction operations needed to realise a project including, amongst others, a high level description of the activities, data requirements, resource requirements, ordering relations and temporal constraints, time-duration, abstraction, cost etc. This information is generated and manipulated at different process stages throughout a construction project's lifecycle by the various participants employing different software applications. Each of these applications focuses on particular aspects of the process For insistence CAD design applications specify the project products and parts to be realised by construction or production processes hence provide the design data requirements of processes, cost estimating defines the processes needed to realise the designed project products, resources needed to perform the processes and determine the cost of the activities based on the cost rates of the resources and planning and scheduling specifies time of occurrence, duration, sequence, resource requirements etc. of each process and constraints that may exists between them. All this information is fragmented and yet has to be shared by the different construction process stages and supporting software applications. While "Fragmented information" refers to the process information generated and represented at the various process stages using different supporting software applications with varying terminology and file formats. "Shared information" presents the information exchanged or shared, by various applications within the pre-implementation of construction processes on site.

6.2.3 Construction Computer applications

The advent of Information Technology (IT) has presented promising opportunities for the development of digital applications that may significantly improve the traditional management of projects information in the construction domain. To date, there are a wide range of computer applications employed across construction process stages, to support the description and representation of construction processes. Some of the primary areas in which computer applications are widely used to support the description and management of construction processes are the pre-construction process stages These include design; cost estimating, and project scheduling applications areas. Some of the software applications that support these areas include, AutoCAD for design, Construction Computer Software (CCS) for cost estimating and Microsoft project for planning and scheduling, each of which is developed and supported by different programming and control languages. Each of these applications focuses on particular aspects of a construction process and uses a unique representation and terminology to describe the processes and represent the process information. The "Process information" in this context refers to the information describing the construction processes, in terms of the data types introduced within the different software applications supporting the various construction application areas. For example, a typical construction process description should include design data requirements i.e. design data handled by AutoCAD such as attribute specification of a product to be realised by construction processes; construction processes plan handled by Microsoft Project such as description of tasks, ordering relations, time-duration, constraints etc.; and cost information handled by CCS such as description of tasks, resource requirements and their cost effects.

This shows that each of these applications focus on different aspects of a process to generate and manipulate all the information required during the realisation and management of construction projects and may need to share at-least some of the information. However, almost all of the construction software applications work in a stand-alone manner each with unique representations and terminologies, therefore, do not share the same process concepts even when the same terminology is used, or different terminologies may be used for the same process concept. Hence the meaning of the terminologies representing process information in any of these applications is entirely dependent on the context in which it is viewed and interpreted. As a result information exchange between construction software applications is problematic

6.2.4 Communication

In the construction industry, collaboration between several professionals who usually come from diverse organisations and the interoperability of the numerous computer applications, which are normally introduced throughout the project's lifecycle, to solve specific problems and support information management are cornerstone for the successful achievement of construction projects. Hence, communication and information sharing has become of paramount importance in construction organisations. However, in all types of communication, the ability to share information is often hindered because the meaning of the information can be drastically affected by the context in which it is viewed and interpreted (Cutting, 2000) This is due to the diversity and complexity of construction information handled during the project's life cycle and the barrier to the exchange of the information, which has resulted from the division of work groups and the differences in the representations of information in the systems. Additionally, different construction functions and applications may use different terms to mean the exact same concept or the use exact same term to mean different concepts. This has resulted in misinterpretation of the contents of the information exchanged between applications and poor communication practice in the sector.

Although, the use of computer applications is wide spread in construction today, most information exchange and sharing still has to be done in some form of manual transaction. The reason is that, most of the construction computer applications are used throughout project's lifecycle to deal with individual problems and support specific information requirements working in a stand-alone manner each with its own and unique information representation; file formats and terminologies. Hence, information sharing between applications is inconsistent, difficult and some times impossible without the application of specific translators, as the meaning of the information can be completely dependent on the context in which it is interpreted. One of the primary reasons for this problem is the lack of a common language for specifying construction processes.

6.2.5 The challenges

Traditionally information exchange and communications has been poor and inconsistent in the construction industry. In order to reuse information generated at one process stage using a specific application as an input to another application, manual methods of exchange are still sometimes used. Finding an effective way of communication and information sharing has become the greatest challenge faced by construction organisations today. As a result the capability of AEC/FM applications to inter-operate has become a major issue for academics, researchers and standardisation bodies As a result there have been efforts in developing interoperability mechanisms and standards for information exchange between AEC/FM software applications. Several data standards have been developed for product description and exchange of construction projects (IAI 1997 & 1998; ISO/TC184/SC4, 1993a). However, among these efforts towards interoperability the process centred view of the construction information has received little attention.

The first stage in developing mechanisms for interoperability of process related construction computer applications lies in the development of a common specification of construction processes. This requires a neutral language that would facilitate the specification of construction processes. Developing a language for the specification of construction processes is one of the challenging task for researchers and standardisation bodies. One of the objectives of this research study is, to investigate the applicability of the manufacturing Process Specification Language (PSL) for the specification of construction processes.

6.3 PSL from Construction Processes Perspective

PSL was mainly developed and used in the manufacturing sector for the specification and representation of manufacturing processes information to integrate multiple process related manufacturing applications. This project proceeded to investigate the applicability of this language in construction by focusing on the application areas within the pre-construction process stages that PSL is expected to support the specification and representation of construction process information and to assess how the process concepts in these areas impact on the use of PSL in construction environment.

This section, first briefly looks at the application areas within the pre-construction process stages chosen as an investigation ground for the applicability of PSL in construction, and then identifies the process concepts intrinsic to theses application areas and discusses the process specification and representation needs for these areas. The application areas are reviewed and process related information representations are identified through technical analysis of the software applications supporting these areas. The data types used in these software applications that are necessary for representing the construction processes information within the pre-construction process stages were analyzed and compared with the manufacturing process concepts defined in the PSL ontology. The PSL ontology is assessed to identify whether it contains concepts and that can capture the intuitive meaning of the construction process specification requirements or data types of the design, cost estimating and scheduling applications. The expectation is that either PSL will need to incorporate definitions for the construction process specification requirements (data types) or these requirements will need to be defined within new extensions to the PSL ontology.

The following sections will identify and present the construction process representation requirements that are intrinsic to the pre-construction process stages. The manufacturing process concepts of the PSL ontology were assessed to determine the applicability of the PSL language for process related architectural design, cost estimating and scheduling information representations' semantic concepts definitions and based on the result of the investigation conclusions were drawn on the applicability of the language in the construction environment.

6.3.1 Process Specification Requirements in the Pre-Construction stages

The pre-construction process stages are the areas in which computer applications are widely used to support the description, and planning of construction processes. The requirements related to the pre-construction process stages that are necessary for representing construction processes can be identified through analysis of the software applications supporting these stages. This section briefly looks at the architectural design, cost estimating and scheduling application areas of the pre-construction process stages and the requirements are gathered through identification of the data types that are used, in these software applications as a method of description and representation of construction processes. Programs supporting the architectural design, cost estimating and scheduling applications areas which are the focus of this research study are: AutoCAD, Construction Computer Software (CCS) and Microsoft project. These applications describe construction processes from design or product data requirement, cost effects and scheduling and planning point of view and were analyzed to identify, what process related requirements need to be represented in each of these applications AutoCAD is widely known as a product related application, however in this research study it is analysed as a process related application because it deals with project information representation including process related data. These packages were selected as test applications for the investigation of the applicability of PSL in construction, because AutoCAD and Microsoft Project application are currently the most widely used applications in the UK construction organizations, and CCS is an application with a complex information representation structure.

As the one of the main objectives of this research study is on information exchange between the three software applications, the process specification requirements that represent shared process concepts between the applications are the focus of investigation. The construction process specification requirements that represent shared process concepts are tabulated together against the PSL process concepts in the following sections in order to summarize the result of the investigation of the applicability of PSL in construction. The process specification requirements (or process information representations) related to architectural design, cost estimating and scheduling and planning application areas that are supported by AutoCAD, CCS and Microsoft project applications are described as follows

6.3.1.1 *Process Specification Requirements related Architectural Design*

The AutoCAD drawing objects provide a virtual representation of the architectural products in a project, which are being realised by construction processes. AutoCAD provides geometrical representation and textual specification of the project objects The geometrical representations may convey information such as dimensions and location of products in a project. However realization of the project products require more information than that can be conveyed through the drawing information representations of AutoCAD application. AutoCAD drawing objects are the main entities by which a designer captures a compound (composite) and component (detailed) objects or products in a project. Textual attributes can be attached to each drawing object to represent more detailed descriptions based on its purpose in the construction process for the product represented by the drawing object. **Figure 6.1** illustrates the basic data model for AutoCAD application, which is developed based on the analysis carried out to identify the process related project information representations of AutoCAD and other applications such as database and spreadsheets that can be used for structuring textual information of drawing objects. Attributes can be extracted from AutoCAD drawing objects and used in a spreadsheet or database to produce textual information. Database file formats (spreadsheet files) are widely used in applications such as AutoCAD for structuring information extracted from drawing objects, however there are no default data fields set for this purpose. AutoCAD application users or designers need to define the data fields for data types and relationships between them that need to represent the process related information of a project at the design stage.

Figure 6.1 Shows the AutoCAD design application's basic project information representation entities. The basic entities are drawing objects, blocks, and attributes that can be associated with each drawing and block for textual characterisation of the drawing objects.

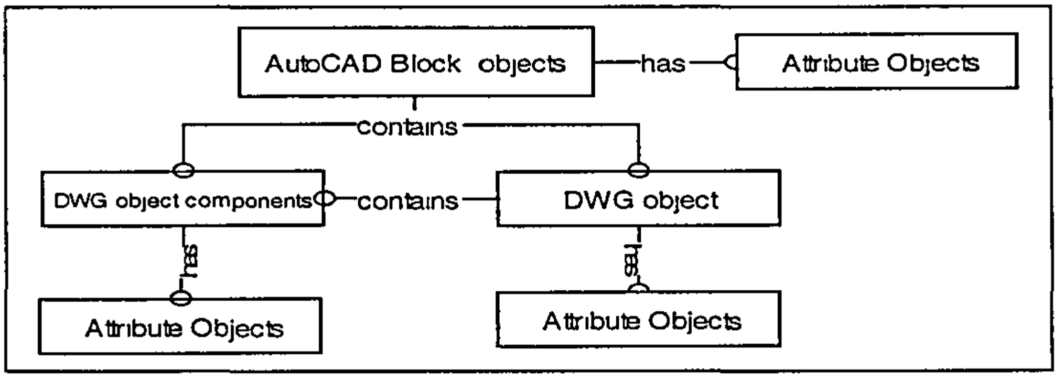


Figure 6. 1. Design application model based on AutoCAD

The process related information that need to be represented in architectural design stage are: the name of the geometrical (i.e. the drawing objects) representation of the project products; and the textual attributes that can be attached to and extracted from each drawing object and structured in database file formats to provide information related to the project product's construction processes. For further clarification of the design information representation data types a glossary of technical terms for the AutoCAD software application is attached in **Appendix A.** The following provides a brief description of the AutoCAD information representations depicted in **figure 6.1**·

DWG object: geometrical representation of project products, which is being realized/constructed by construction processes. This representation provides the overall graphical/geometrical view of the compound and component project products and some data that can be attached to the drawings this includes dimensions, orientation and locations of the overall object and parts.

Block objects: these are AutoCAD symbols representing drawing objects simple geometric shapes and complex objects A *block* is a group of entities or drawings or objects that can be manipulated as a single unit and any instance of a block can be exploded into detailed entities or drawing objects.

Attribute objects: Attribute objects are textual definition of characterises of the block or drawing objects.

Spreadsheet files: AutoCAD allows designers to extract attribute information from drawing and block objects as a list in CDF (comma delimited format), SDF (space delimited format) or DXF (data exchange format) for use in other programs such as spreadsheet files to produce information such as parts list, bill of materials (BOM), or bill of quantities (BoQ). The CDF, SDF and DXF file formats can be imported into any word processing program that accepts ASCII files, these include, Microsoft Excel and Microsoft Access files.

6.3.1.2 *Process Specification Requirements related Cost Estimating*

In this section the requirements for the representation of process information related to cost estimating are identified through analysis of the cost estimating methods used in the CCS application. **Figure 6.2** shows the CCS cost estimating applications basic model developed using the applications method of cost estimating. The model illustrates the basic cost estimating entities and the data types associated with each basic entity i e. the fields associated with the basic entities. The main entities are: CCS macros, operations and resources Macros are products of projects or the high level or complex construction operations. Macros can be found at the different level of abstraction. Operations are activities at the lowest level of decomposition of the macros that are performed to achieve the next higher-level macros. Finally the third basic entity is resources (labour, plant, matenal, temporary matenal, subcontractors etc.) that perform the operations All these entities have numerous data fields associated with them.

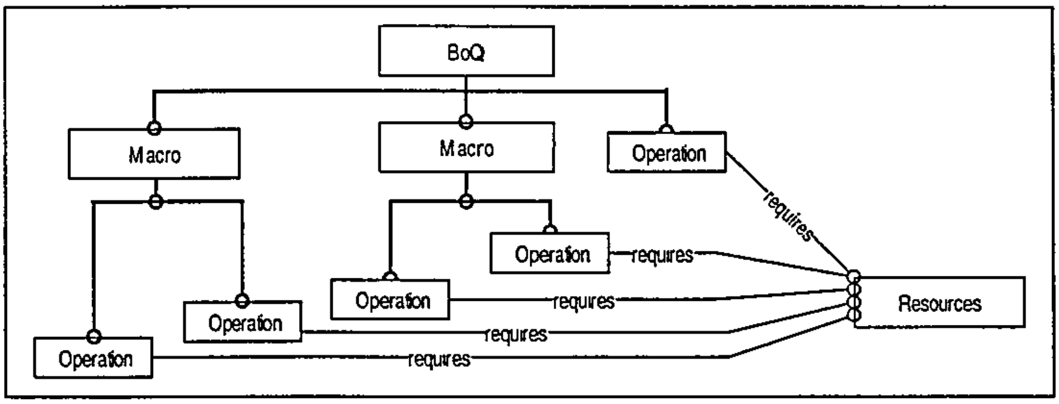


Figure 6 2. Cost estimating application model based on CCS

The main entities and the associated attnbutes represent the construction process specification requirements. For further clanfication of the cost estimating information representation data types a glossary of technical terms for the CCS software application is attached in **Appendix A.** A large number of attnbutes are associated with these basic entities. In this case, Construction Computer Software (CCS) has the following data structures:

- Macros have the following data fields associated with each macro operation
  - CCS Page Number, Item Number, Op Code (Operation Code), Op Code Descnption,
  - Op Code Unit, Billed Quantity, Nett Rate, Nett Amount
  - Trade code
  - Operations
  - Nett Split Rates
  - Op Code Attnbutes (Attr)
  - Page / Item: page number and item number combined
  - Gross Rate, Billed Gross Amount
  - Selling Rate, Billed Selling Amount

- □ Operations: have the following data fields associated with each operation
  - Item Number, Op Code (Operation Code), Op Code Description, Op Code Unit, Quantity, Nett Rate, Nett Amount
  - Work sheet resources and cost rates
  - Nett Split Rates, Op Code Attributes (Attr)
  - Page / Item, Gross Rate, Billed Gross Amount
  - Selling Rate, Billed Selling Amount
- □ Resources: have the following data fields associated with each resource
  - Resource Type, Resource Code, Resource Description, Final Rate, Resource Unit, Resource Attributes (Attr), Base Rate
  - Resource group codes
  - Resource class codes, Production Factor, Production Code
  - Usage, Usage Value
  - Resource amount

6.3.1.3 *Process Specification Requirements related Scheduling and Planning*

One of the computer applications widely supporting construction planning and scheduling is Microsoft Project (Microsoft INC.) **Figure 6.3** (Thomas Froses, 1997) illustrates the basic data model adopted by Primavera Project Planner (Primavera 1991) and Microsoft Project (Microsoft 1994). The model is centered around the *activity* or *task* (essentially a process entity), other main entities are the *project* itself, *resources*, *resource utilization* (the allocation of specific resources to specific activities), *precedence logic* (the inter-activity sequencing constraints), and *work calendars* (descriptions of the length and timing of the work week available for different types of processes).

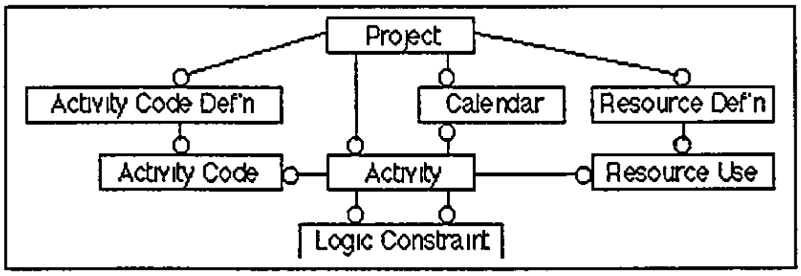


Figure 6 3. Scheduling application model based on Microsoft Project planner (source: Thomas Froses 1997)

A large number of attributes are associated with these basic entities. The basic entities and the associated attributes represent the process information specification requirements related scheduling. For further clarification of the application's information representation data types a glossary of Microsoft Project application basic entities and the associated data types (technical terms) is attached in **Appendix A**. In this case, Microsoft Project, has the following data structures.

- Tasks have the following data fields associated with them, including other data related to these:
  - durations, start and finish dates (early and late), float or slack time, and task delays,
  - work hours, task progress, and earned value analysis,
  - predecessor, successor, and scheduling constraints,
  - task costs,
  - hierarchical breakdowns (outlines), summary and rollup schedules, subprojects, and work breakdown structures,
  - resources,
  - task ID's, names, user-defined text and flag fields, notes, contact names, creation dates, and OLE linked objects,
  - milestone, criticality, priority, confirmation, and update indicators,
  - multiple versions of many of the above data items are stored, including *normal*, *actual*, *baseline*, *remaining*, and numerous user-defined versions.
- resources have the following data fields associated with them, including other data related to these:
  - costs, work hours, work calendars, variances, accrual basis, overtime, rates,
  - resource ID's, names, groups, units, notes and OLE linked objects,
  - progress, peak use, and over-allocation indicators,
  - multiple versions of many of the above data items are stored, including *normal*, actual, baseline, remaining, and overtime.

6.4 Construction Process Concepts Analysis and Mapping to PSL ontology

6.4.1 Introduction

In the previous sections (sections 6.3.1.1-6.3.1.3) sets of data types (process specification requirements) that are necessary for representing construction process information within the pre-construction process stages (design, cost estimating, and scheduling or planning) were identified and gathered through analysis of AutoCAD, CCS, and Microsoft project scheduling applications selected to support these stages. In order to facilitate the mapping of the process specification requirements of these stages to the PSL ontology, this section, first presents process related design, cost estimating, and scheduling or planning concepts maps as shown in **figures 6.5, 6.6 & 6.7**, thorough conceptual analysis and mapping of the applications' data types. These maps provided a framework within which a theory that postulates relationships between the data types might be stated. Based on these maps the construction process concepts related to the design, cost estimating, and scheduling applications are defined using natural language statements of the applications data types' semantic concepts The PSL ontology also is a map of manufacturing process concepts developed through conceptual analysis and mapping of the process representations related to various software packages and application areas in the manufacturing environment. The manufacturing process concepts in the PSL ontology are defined formally using Knowledge Interchange Format (KIF) specifications. Based on analysis of the natural language statements of the applications data types' semantic concepts and the manufacturing process concepts defined in PSL ontology, the applications construction process representation data types shown in **figures 6.5, 6.6 & 6.7** were mapped to the PSL ontology shown in **figure 6.4**, and the results of the mapping process is shown in **figures 6.8 to 6.9** and summarized in **table 6.1.** Finally, analysis of the mapping process is presented followed by conclusions. In the following sections a brief review of the PSL ontology is presented and the construction process related design, cost estimating and scheduling concepts maps are described.

6.4.2 PSL Ontology Concepts and Current Theories and Extensions

The Process Specification language is a neutral language, for the representation of process information, related to manufacturing, including all processes in the design/manufacturing life cycle. This interchange language is unique due to the formal semantic definitions (the ontology) that underlie the language. All concepts with the PSL representations are unambiguously and formally defined in the PSL Ontology. PSL Ontology is a set of specialised terminology along with some specification of the meaning of the terms in the lexicon. It provides rigorous and unambiguous definitions of the concepts necessary for specifying manufacturing processes to enable the exchange of process information. The PSL ontology is represented using the KIF specification

(Genesereth, and Fikes, 1992). KIF is a formal language based on first-order logic developed for the exchange of knowledge among different computer programs with disparate representations KIF provides the level of rigor necessary to define concepts in the ontology unambiguously, a necessary characteristic to exchange manufacturing process information using the PSL ontology.

The foundation of the ontology is a set of process-related concepts that constitute the core of the PSL ontology and set of extensions to these core concepts to ensure the robustness of the ontology. The PSL ontology is organized modularly with a set of core concepts and multiple extensions add to the core. The PSL-Core is the set of axioms written in KIF (the Knowledge Interchange Format) and using only the nonlogical lexicon of PSL Core. The extensions form a lattice of extensions to PSL-Core. PSL-Core is not strong enough to provide definitions of the many auxiliary notions that become necessary to describe all intuitions about manufacturing processes. To supplement the concepts of PSL Core, the ontology includes a set of extensions that introduce new terminology. A PSL extension provides the logical expressiveness to express information involving concepts that are not explicitly specified in the PSL Core. A distinction is drawn between definitional and nondefinitional extensions of the PSL Core. A definitional extension is an extension whose nonlogical lexicon can be completely defined in terms of the PSL-Core. Definitional extensions add no new expressive power to PSL-Core. Nondefinitional extensions (also called core theories) are extensions of the PSL Core that involve at least one new primitive not contained in the PSL Core. The architecture of PSL ontology shown in **Figure 6.4** presents the possible parts of PSL that are suggested and selected to be applicable and provide semantic concepts definitions for AutoCAD design, CCS cost estimating and Microsoft project scheduling applications' data types. In the following sections a review of the PSL parts and their concepts are presented. Detailed formal and informal definitions of the PSL ontology concepts and parts are available in (http://www.mel.nist.gov/psl/ontology.html).

6.4.2.1 Core Theories of PSL Ontology

PSL-Core:

- **PSL-Core:** The purpose of PSL-Core is to axiomatize a set of intuitive semantic primitives that is adequate for describing the fundamental concepts of manufacturing processes.

Outer Core

- **Subactivity Theory:** The PSL Ontology uses the sub-activity relation to capture the basic intuitions for the composition of activities.
- **Theory of Occurrence Trees:** An occurrence tree is the set of all discrete sequences of activity occurrences.
- **Theory of Discrete States:** The Discrete State core theory is intended to capture the basic intuitions about states and their relationship to activities.
- **Theory of Atomic Activities: The** core theory of Atomic Activities provides axioms for intuitions about the concurrent aggregation of primitive activities.
- **Theory of Complex Activities:** This core theory provides the foundation for representing and reasoning about complex activities and the relationship between occurrences of an activity and occurrences of its subactivities. Occurrences of complex activities correspond to sets of occurrences of subactivities; in particular, these sets are subtrees of the occurrence tree
- **Activity Occurrence:** The axioms of the Activity Occurrences core theory ensure that complex activity occurrences correspond to branches of activity trees. A subactivity occurrence corresponds to a sub-branch of the branch corresponding to the complex activity occurrence.

Duration and Ordering Theories

- **Duration Theory:** The axioms of duration core theory ensure that duration relationships between time point and duration and between durations.
- **Subactivity Occurrence Ordering:** The axioms of this core theory defines the relations of precedes, root, leaf, next, for activity occurrences.
- **Iterated Activities:** The axioms of this core theory define the concepts of repeating occurrence of activities.
- **Occurrence Tree Automorphisms:** There is no clear statement about the use of this core theory.
- **Envelopes and Umbrae:** There is no clear statement about the use of this core theory.

Resource Theories

- **Resource Requirements Theory:** defines the concepts of resource that is any object, which is required by some activity.
- **Resource Sets.** underdevelopment

Actor and Agent Theories

- **Activity Performance:** underdevelopment

6.4.2.2 Definitional Extensions of PSL

Activity Extensions

Branch Structure

- Deterministic Activities: Permuting Branch Structure
- Nondeterministic Activities: Folding Branch Structure
- Nondeterministic Activities: Branch Structure and Ordering
- Nondeterministic Activities: Repetitive Branch Structure

Spectrum

- Spectrum of Activities: Permuting Activity Trees
- Spectrum of Activities: Compacting Branch Structure
- Spectrum of Activities: Activity Trees and Reordering
- Spectrum and Subtree Containment

Embedded Activity Trees

- Embedding Constraints for Activities
- Skeletal Activity Trees

Atomic Activities

- Atomic Activities: Upwards Concurrency
- Atomic Activities: Downwards Concurrency
- Spectrum for Atomic Activities

Temporal and State Extensions

Preconditions

- Preconditions for Activities
- State-based Preconditions for Activities
- Time-based Preconditions for Activities
- Preconditions based on State and Time
- Occurrence-based Preconditions for Activities
- Preventable Preconditions for Activities
- Periodic Preconditions for Activities
- Spoilage Preconditions for Activities

Effects

- Effects of Activities
- State-based Effects of Activities
- Time-based Effects of Activities
- Occurrence-based Effects of Activities
- Effects of Activities. State and Occurrences
- Effects of Activities: Time and Occurrences
- Fluent Trees

Distribution

- Distribution of Complex Activities
- State-based Distribution of Complex Activities
- Time-based Distribution of Complex Activities
- Distribution based on State and Time

Variation

- Variation of Complex Activities
- State-based Variation of Complex Activities
- Time-based Variation of Complex Activities
- Variation based on State and Time

Embedded Activity Trees

- Embedded Activities: Plans
- Embedded Activities: Temporal Spread

Atomic Activities

- Variation of Atomic Activities: Upward Concurrency
- Variation of Atomic Activities: Downward Concurrency
- Interfering Preconditions
- Clobbering Effects
- Variation of Interfering Preconditions
- Variation of Clobbering Effects

Activity Ordering and Duration Extensions

Duration

- Duration Constraints for Activity Occurrences
- State-based Duration
- Time-based Duration
- Duration based on State and Time
- Duration-based Effects
- Effects of Activities based on Duration and State
- Ordering and Duration Constraints on Activity Occurrences
- Ordering and Duration Constraints on Embedded Activity Occurrences
- Spoilage Preconditions for Activities
- Scheduled Embedding Constraints

Subactivity Occurrence Orderings

- Strong Partially Ordered Activities
- Weak Partially Ordered Activities
- Complex Activity Occurrence Orderings

Envelopes and Umbrae

- Interruptable Activities

Iterated Occurrence Orderings

- Classes of Iterated Activity Occurrences

Resource Roles

- Resource Roles
- Capacity-Based Concurrency
- Reasoning About Resource Divisibility
- Reasoning about Resource Usage

Resource Sets

- Resource Set-Based Activities
- Substitutable Resources
- Homogenous Resource Sets
- Inventory Resource Sets
- Resource Pools

Processor Activity Extensions

- Processor Activities
- Resource Paths

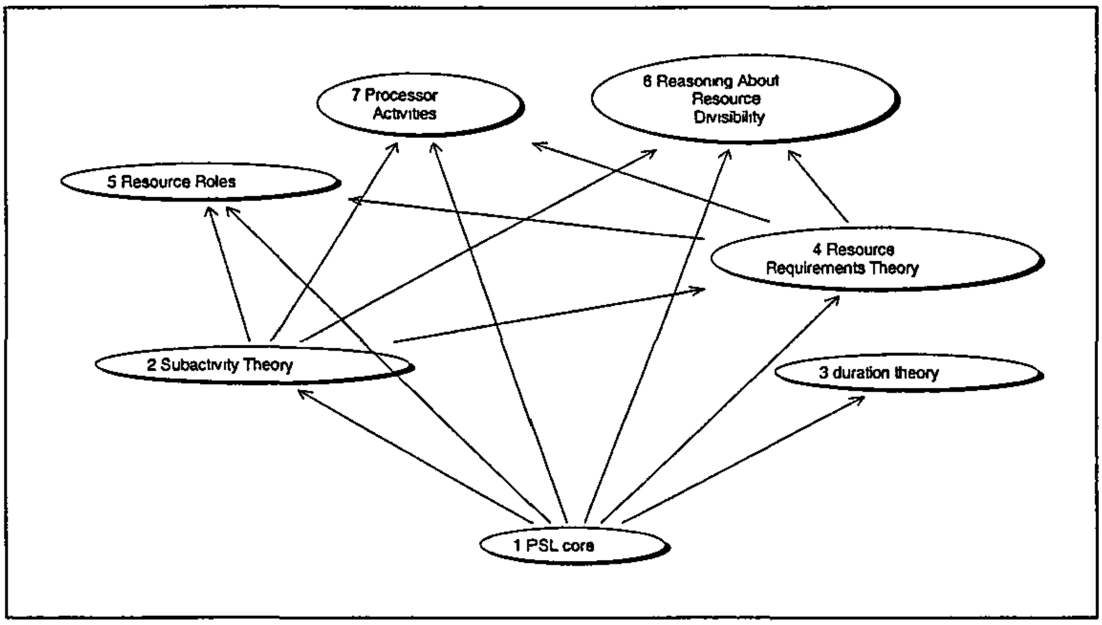


Figure 6.4. PSL ontology (manufactunng process Concepts Map)

6.4.3 *Process Information Concepts within the Pre-Construction Stages*

AutoCAD Design Related Construction Process Concepts

From the data types; drawing objects (or project product design element), construction method, and matenal of construction of **figure 6.5** (conceptual map for AutoCAD application) the design related construction process concepts are defined using natural language statements as follows "*In AutoCAD design, drawing objects (project design elements) that represent construction or project products are made up of other objects or component parts, that are constructed by the construction method (construction activities) from materials that are specified by the material of construction for the component parts. Hence the construction activity transform, modify, or destroy the material in the development of the component parts and eventually produce the target product*".

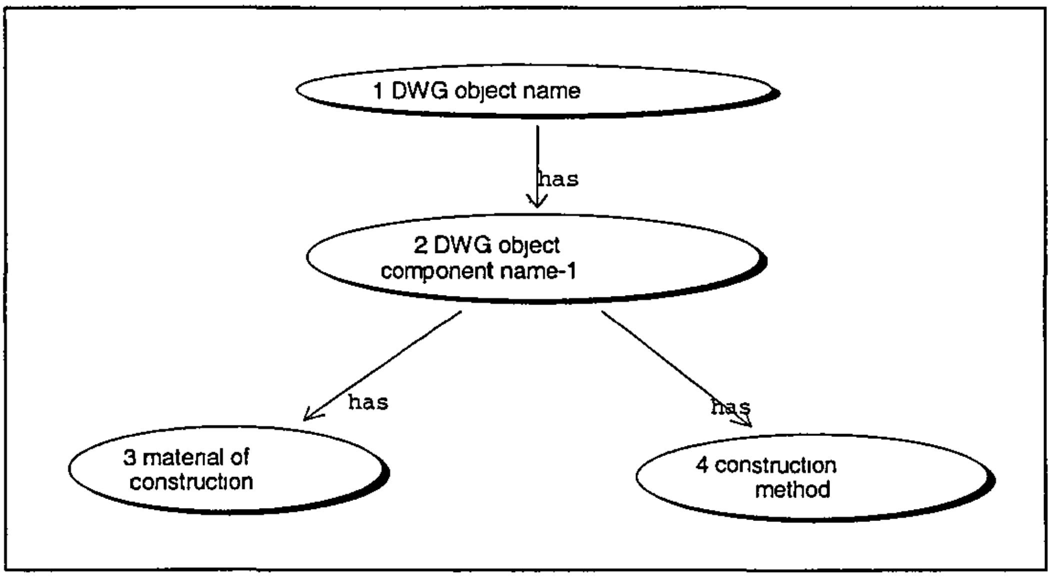


Figure 6 5. AutoCAD design data types map

CCS Cost Estimating Related Construction Process Concepts definition

The conceptual map (conceptual models) for CCS application as shown in **figures 6.6 (a & b)** are developed into parts. This is because CCS unlike the other two candidate applications use different data formats for the input files and output or report files. The first is cost estimating concepts map as shown in **figure 6.6 a,** which is developed from data types of the applications input files. The second is cost concepts map as shown in figure **6.6 b,** which is developed from data types of the applications output or report files.

From the data types of the CCS input files; bill items, billed qty, Nett-rates & Nett-amounts of bill items, resource categories (plant, material, labour, temporary materials, subcontracts etc.) and resource rates of **figure 6.6 (a and b)** (conceptual map for CCS application input file) the cost estimating related construction process concepts are defined suing a natural language statements as follows: "*in CCS cost estimating bill items (that are assigned with a macro or simple operation code) are construction project products to be constructed or construction activities to be performed, that could be complex activities that need to be accomplished by other set of simple activities or a simple activity that can not decompose into other activities. The construction project product is constructed by set of simple and/or complex activities. Complex activities may decompose into set of simple activities, which require resources categorised into plant,* *material, labour, temporary material, subcontract etc. and these activities use, transform, modify or destroy resources. These resources have cost rates based on their utilisation by activities, which are used to calculate the Nett-rate for the simple operation or activity and the Nett-rates for the simple operation or activities contribute to the Nett-rate of the operation at the next higher level (the macros). The project product, the complex and simple operations have billed quantities and this quantity with the Nett-rate generates the Nett-amount of the projects products and operations".*

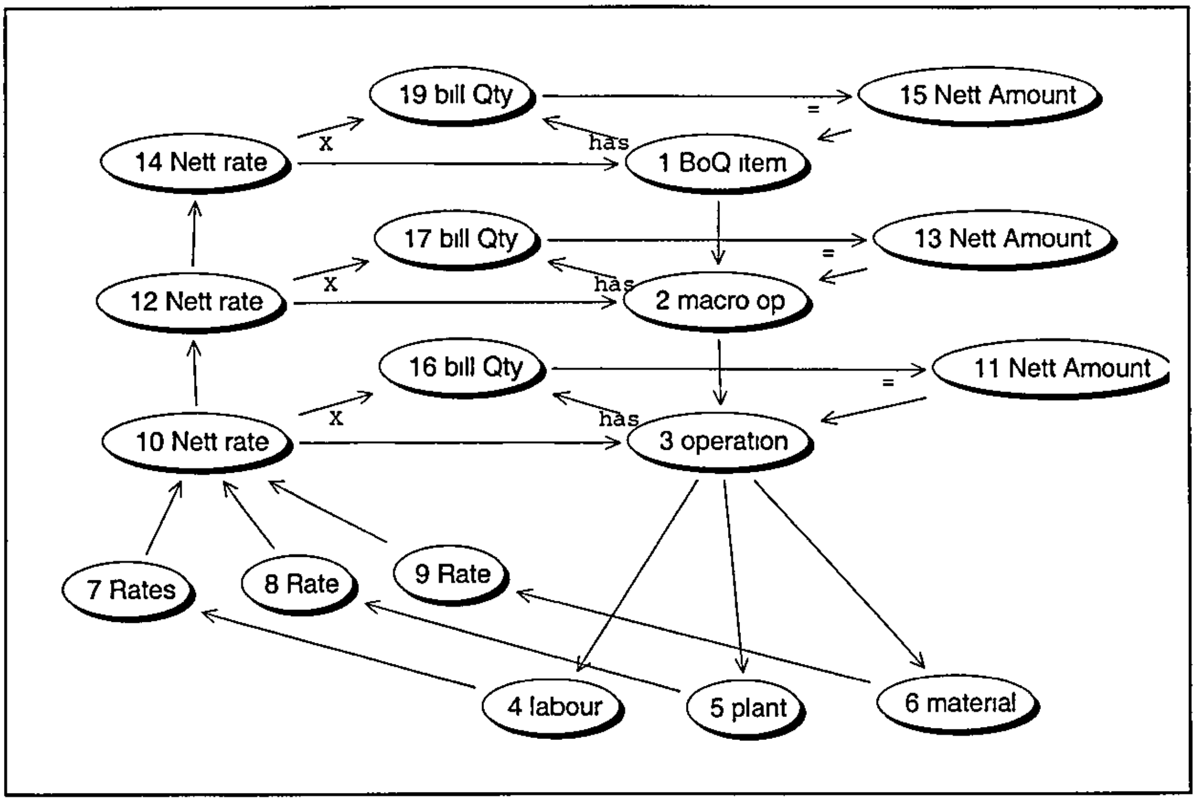


Figure 6.6 (a) CCS Input cost estimating data types map

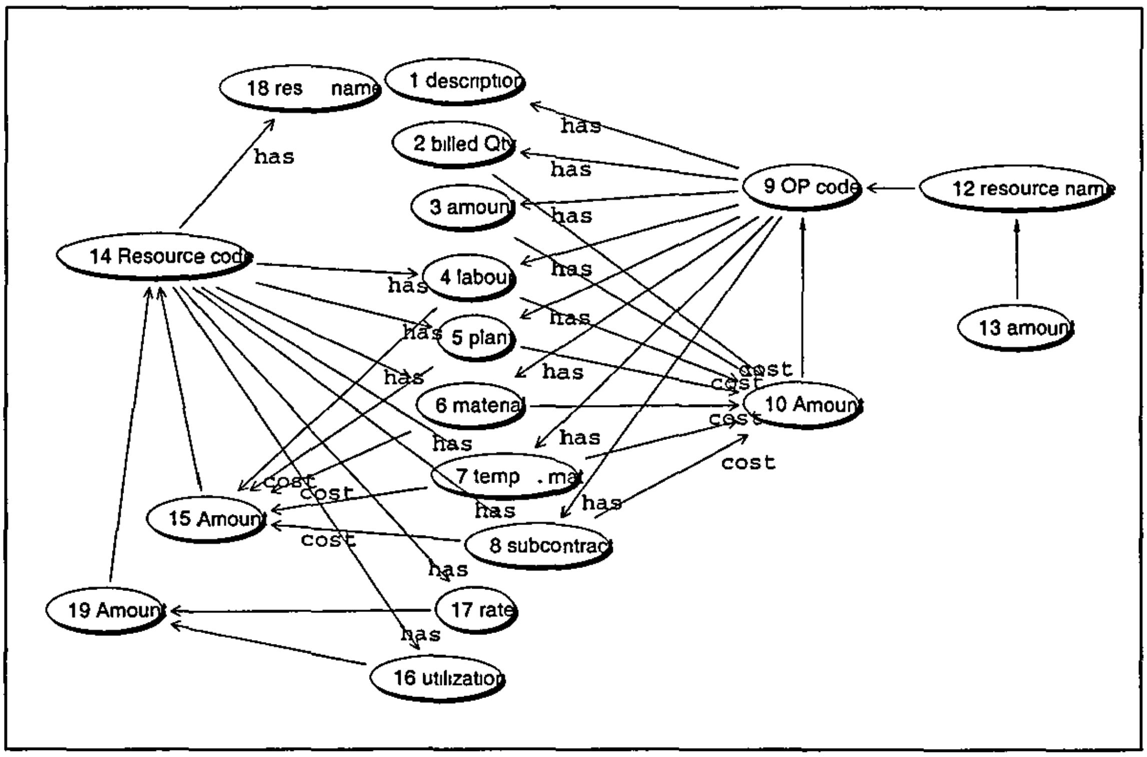


Figure 6.6 (b) CCS output Cost data types map

Microsoft Project Scheduling Related Construction Process Concepts definition

From the data types of the Microsoft project scheduling application; summary task and task (sub tasks), resources, cost/use and standard rates of resources, total and fixed costs of summary tasks and tasks of **figure 6.7** (the conceptual map for Microsoft project scheduling application files) the scheduling and planning related construction process concepts are defined using a natural language statements as follows: " *in scheduling summary tasks have sub tasks that require resources which have cost/use (cost of resources per use by a task) or standard rate(cost rate per duration of a task). Tasks transform, destroy, modify or use resources and a task can have a fixed cost (e.g. subcontract's cost) or cost caused by resources utilizations While resources with cost/use are assigned to a task the cost/use of all the resources add up to produce a total cost for the task, the resources with standard rates require duration of the task to produce the total cost of the task.* "

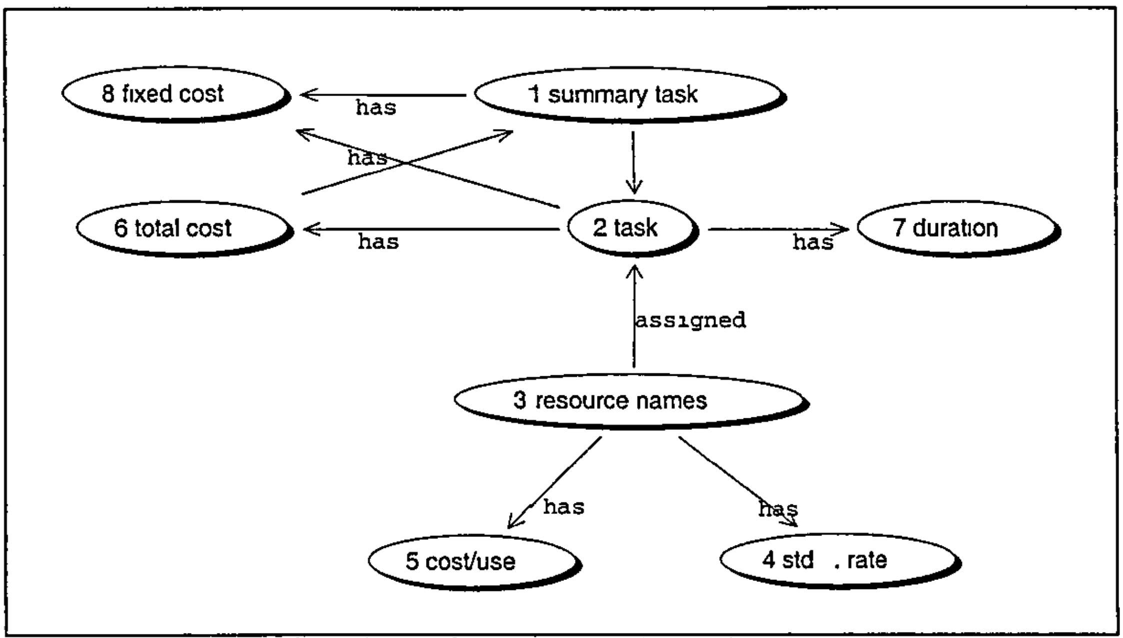


Figure 6.7 Microsoft Project scheduling data types map

6.5 Analysis of the Construction Process Representations and Mapping to PSL Ontology

The objective of this section is to assess the applicability of PSL in the pre-construction process stages through multidimensional conceptual analysis between the AutoCAD, CCS and Microsoft project scheduling applications data types (represented by construction process related design, cost estimating, and scheduling maps) and PSL ontology, and mapping of the applications data types to the PSL ontology. The implication is that PSL is expected to include definitions for the construction process related design, cost estimating and scheduling data types or new extensions should be developed in such way that construction semantic concepts that can define the applications' data types could be added to the PSL ontology. The construction process representations within the pre-construction process stages (shown in **figures 6.5 to 6.7)** were mapped to the PSL ontology (shown in **figures 6.4)**, in order to assess the degree of applicability of the PSL language in the pre-construction process stages.

Mapping of the Applications Data Types to Parts of PSL Ontology

As can be seen from the mapping of the applications' data types to the PSL ontology (see **figure 6.8 to 6.10)**, the applications data types created pointers (shown as arrows pointing to the PSL ontology parts) to the PSL parts when there are concepts in these

PSL parts those are applicable to define the data types, or created pointers to the "NEW COCEPTS", to indicate the need for new extensions to the PSL ontology. The different pointers to the new extensions provide the information required for the development of the new extensions to the PSL ontology by directing the applications data types and existing PSL parts that need to contribute to the definition of the new concepts. For example referring to **figure 6.8,** (CCS cost estimating application's data type mapping to the PSL ontology), the material, plant, and labour rates are new concepts to PSL ontology, and the ontology needs to create new extensions for these concepts hence the mapping created pointers to the new extension that contain concepts of COST, and because these are resource cost rates, the new extensions for COST requires a resources theory which is already defined in PSL ontology, therefore the mapping created a pointer form the "resources requirements theory" part of PSL ontology to the new extension for COST and these rates are related to other data types such as those that rates that contribute to the amount of operations (see Appendix A for the term "amount" of operation). The mapping of the different applications data types to the PSL ontology shown in **figures 6.8, 6.9 & 6.10** are slightly complicated but this is summarized in table **6.1** to provide a clear view of the mapping process.

Based on multidimensional conceptual analysis between the natural language definitions of the AutoCAD design related construction process concepts and the PSL ontology, and mapping of the AutoCAD design data types (shown in **figure 6.5,**) to the PSL ontology the "Processor Activities" part 46 of PSL ontology was identified to contain definitions for the AutoCAD design related construction process concepts. During the mapping of the contents of AutoCAD design concepts map to the PSL ontology the need for new extensions to the PSL ontology to incorporate the "PROJECT DESGN ELEMETNS OR PROUDCTS " concepts was identified (**figure 8**).

Based on multidimensional conceptual analysis between the natural language definitions of the CCS cost estimating related construction process concepts and the PSL ontology, and mapping of the CCS cost estimating data types (shown in **figure 6.6(a & b)**) to the PSL ontology the "*Sub-Activity Theory*", "Theory of *Resource Requirements*" and "*Processor Activities*" parts of PSL are selected as applicable defining theories for the

CCS process concepts. However, dunng the mappıng of the contents of the CCS cost estimatıng concepts map to the PSL ontology the need for new extensıons to the PSL ontology to ıncorporate the "RESOURCE CATEGORY, BILLED QUNATITY, AND COST CONCPETS" were identıfied (**figure 6.9**).

Based on multıdımensional conceptual analysıs between the natural language defınıtıons of the Mıcrosoft Project Schedulıng related construction process concepts and the PSL ontology, and mapping of the Mıcrosoft Project Schedulıng data types (shown in **figure 6.7**) to the PSL ontology the "*Sub-Activity Theory*", "Theory of *Resource Requırements*" and "*Processor Actıvıtıes*" parts of PSL are selected as applıcable definıng theones for the Mıcrosoft project scheduling process concepts. However, dunng the mappıng of the contents of the schedulıng concepts map to the PSL ontology the need for new extensions to the PSL ontology to ıncorporate the "COST CONCPETS" were ıdentıfied (**figure 6.10**).

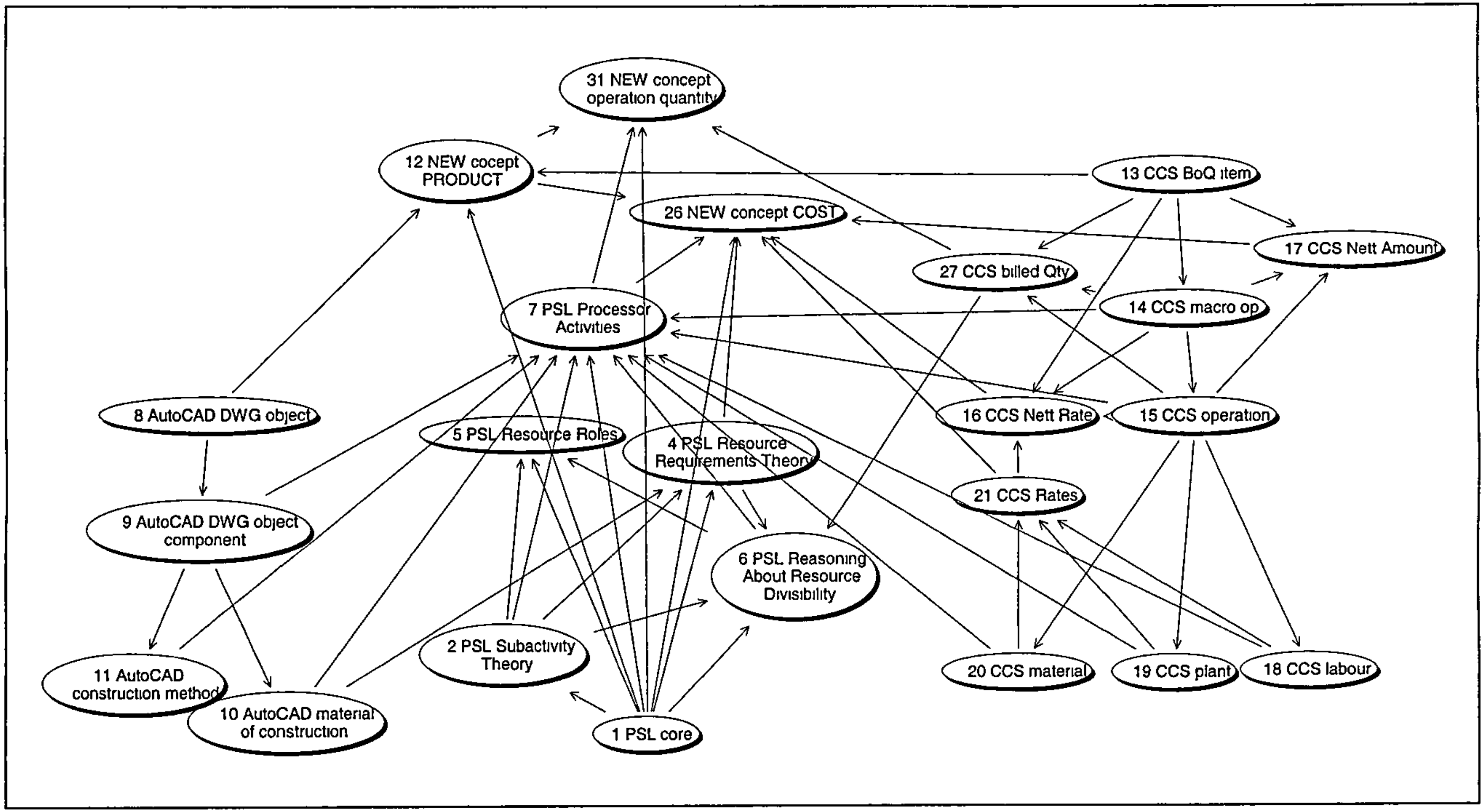


Figure 6.8. Design and cost estimating data types mapping to PSL ontology

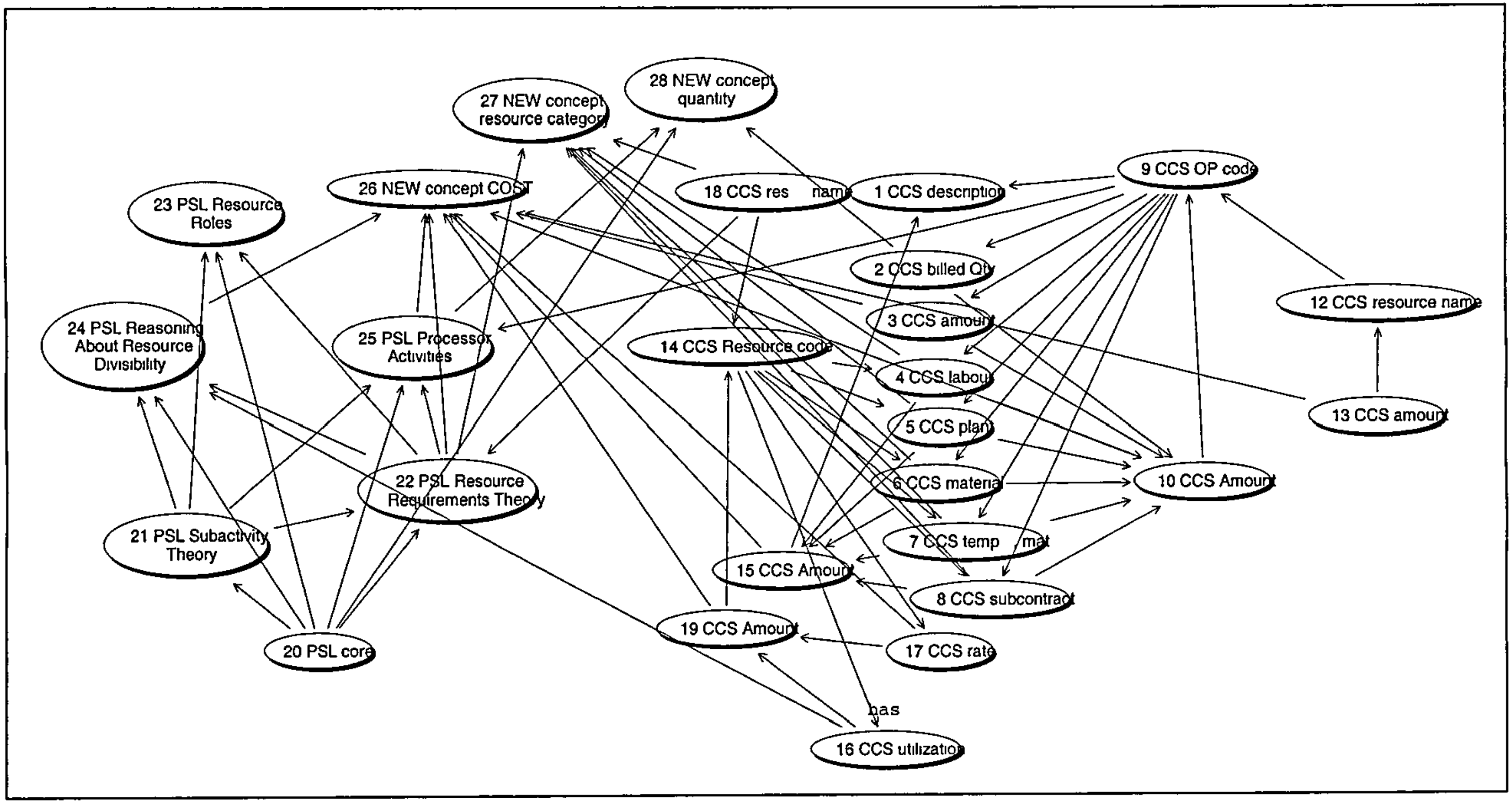


Figure 6.9. CCS Cost data types mapping to PSL ontology

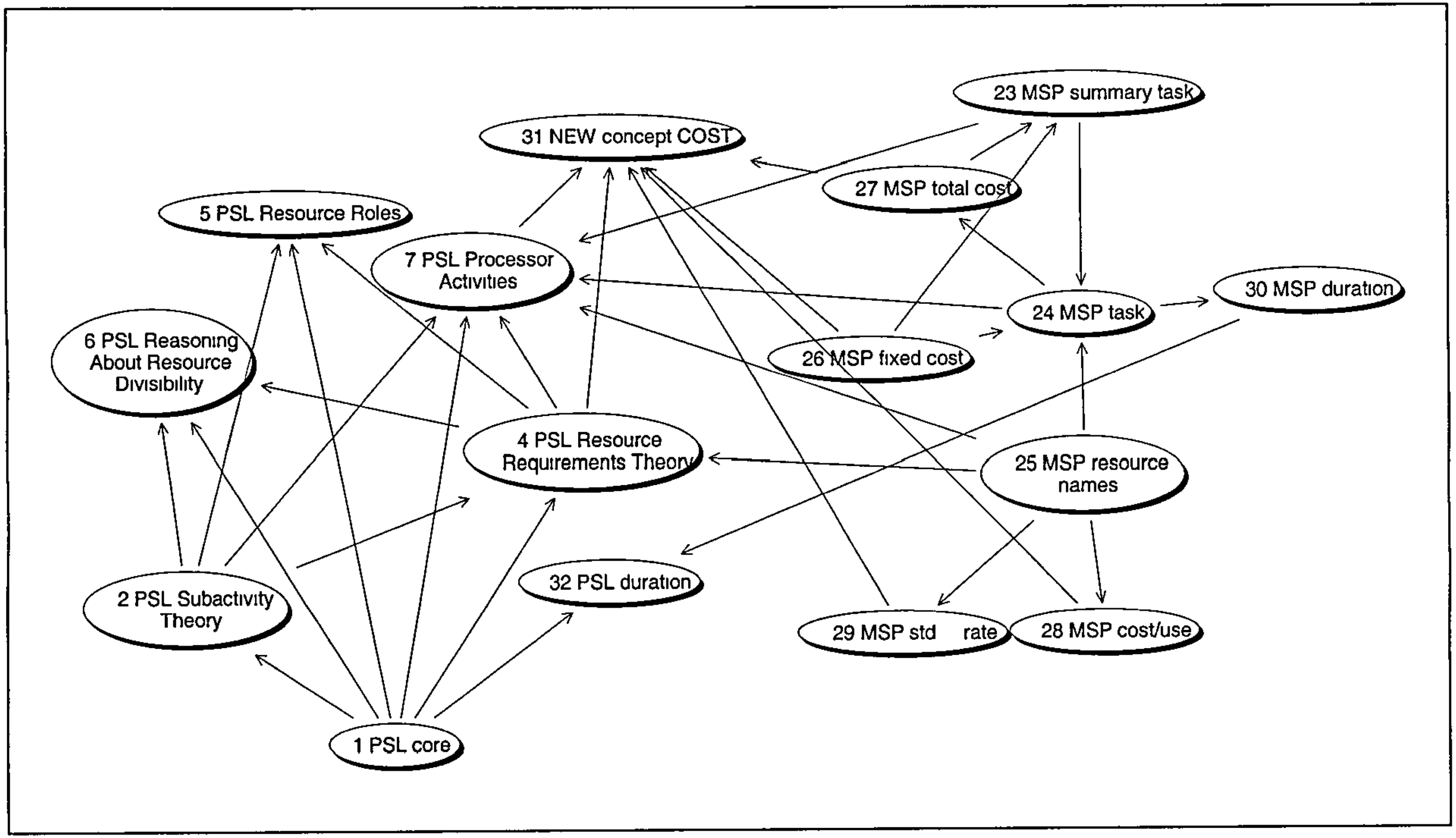


Figure 6.10 Microsoft Project scheduling data types to PSL ontology

CCS Cost estimating Concepts	AutoCAD Design Concepts	MS Project Scheduling Concepts	PSL Parts and new Concepts
Bill of Quantity	**Drawing objects**	**Summary Task**	**Summary Task**
□ BoQ item	□ Object name (such as door)	□	□ NEW COCEPT for PRODUCT
□ Nett rate	□	□ ...	□
□ Nett amount	□	□	□
□ Billed quantity	□ Quantity\number	□	□ NEW COCEPT for QUANTITY
Macro	**Components**	**Summary Task**	**Summary Task**
□ Macro	□ Drawing objects Component name (such as doorframe)	□ Task names	□ PSL Processor activity
□ Nett rate	□ ------	□	□
□ Nett amount	□ -----	□	□
□ Billed quantity	□ Quantity\number	□	□ PSL Reasoning about resource divisibility
Operation	**Method of Construction & Material**	**Task names**	**Task names**
□ Operation	□ Method of Construction	□ Task names	□ PSL Processor activity
□ Billed quantity	□	□	□
□ Nett rate	□	□	□
□ Nett amount	□	□	□
□ material	□ Material of Construction	□ Resource	□ PSL Resource requirement theory
Billed Quantities Amount		**Gantt chart cost**	**Gantt chart cost**
□ Op code description		□ Task	□ PSL Processor activity
□ AMOUNT		□ Total task cost	□ NEW COCEPT for COST
□ LABOUR cost		□ Resource cost/use	□ NEW COCEPT for COST
□ PLANT cost		□ Resource cost/use	□ NEW COCEPT for COST
□ MATERIAL cost		□ Resource cost/us	□ NEW COCEPT for COST
□ TEMP MATERIAL cost		□ Resource cost/use	□ NEW COCEPT for COST
□ SUBCONSTRACT cost		□ Resource cost/use	□ NEW COCEPT for COST
Bill Resource Analysis		**Resource sheet entry**	**Resource sheet entry**
□ RESOURCE NAMES		□ Resource name	□ PSL Resource requirement theory
□ Rate/per utilization		□ Standardrate/perduration	□ NEW COCEPT for COST
□ UTILIZATION		□	□ PSL Reasoning about resource divisibility
□ AMOUNT=rate*utilization		□ Cost/use	□ NEW COCEPT for COST
□ LABOUR		□ Resources	□ NEW COCEPT for RES CATEGORY
□ PLANT		□ Resources	□ NEW COCEPT for RES CATEGORY
□ MATERIAL		□ Resources	□ NEW COCEPT for RES CATEGORY
□ TEMP MATERIAL		□ Resources	□ NEW COCEPT for RES CATEGORY
□ SUBCONSTRACT		□ Resources	□ NEW COCEPT for RES CATEGORY
Resource by Bill Item		**Gantt chart entry**	**Gantt chart entry**
□ OP CODE DESCRIPTION QTY RATE		□ Task	□ PSL Processor activity
□ RESOURCE NAME		□ Resource names	□ PSL Resource requirement theory
□ QUANTITY		□	□ PSL Reasoning about resource divisibility
□ RATE		□ Resource rate	□ NEW COCEPT for COST
□ AMOUNT		□ Cost/use	□ NEW COCEPT for COST

Table 6.1. shared construction process specification requirements and corresponding PSL representation

6.6 Conclusion on the applicability of PSL in Construction

From the literature review of the construction environment and manufacturing it is established that the two sectors are similar in many ways. One of the basic converging point for manufacturing and construction is the unit process concepts, that is a process is a generic concept that can be applied to any activity that produces or outputs a product or result, inputs data or materials, has control methodology and resource requirements; and the five components common to all unit processes. The only difference that can exist between these two environments are: the fixed place of production for manufacturing and completely varying places for construction, the products produced by the processes, and the organisations involved in the productions. The uniqueness of the construction project and the varying participant organisations for each unique project has made the construction environment more complex making the need for a common language between the participating organisations even more important.

One of the objectives of this research study is the investigation of the applicability of the PSL language, developed for manufacturing processes specification and process related applications interoperability, in the construction environment. This objective is achieved using multidimensional conceptual analysis and concept mapping methodology between the construction process concepts within the pre-construction process stages and the PSL ontology (definition of the manufacturing process concepts). Based on the analysis of the result of the concepts mapping shown in **figures 6.8, 6.9 & 6.10**, and summarised in **table 6.1** it was concluded that PSL, should be applicable in the construction environment provided that extensions are developed to incorporate construction specific concepts.

From the mapping of the construction process concepts related to the construction process information, which is dealt within the pre-construction process stages, to the PSL ontology, it was established that in order for PSL to support the specification of the construction processes and facilitate the exchange of process information between AutoCAD design, CCS cost estimating, and Microsoft project scheduling packages, there is a need for new extensions to be developed to the PSL ontology to incorporate the concepts of PROJECT DESGN ELEMETNS OR PROUDCTS", "RESOURCE

CATEGORY, BILLED QUNATITY, AND COST". These new concepts will be further elaborated in the discussion section of the implementation of PSL in a construction scenario.

Based on the conclusions on the applicability of PSL in construction, a decision was made to implement the PSL language in a construction scenario within the pre-construction process stages, the results of this trial implementation of PSL in construction are presented in the following chapters.

Chapter 7 A Scenario for Implementation of PSL in Construction

7.1 Introduction

The Process Specification Language (PSL) is a standard language developed at National Institute of Standards and Technology (NIST) for interoperability of process related applications in the manufacturing environment. The initial development of PSL ontology was from a single scenario, the EDAPS (Electromechanical Design and Planning System) scenario developed by Steve Smith at the University of Maryland (Smith, at al , 1996). After the initial development of PSL, in that scenario there were multiple pilot implementations of the language planned each focusing on different fields/applications within the manufacturing environment, to further expand the language to incorporate the process concepts introduced in various manufacturing application when PSL was used to exchange process information among the applications.

During the various implementations of PSL in the manufacturing environment it has been helpful to engage in scenario analysis (Asia et al. 1994, Kazman et al 1996). Scenario analysis has been defined in a software engineering context as "the process of understanding, analyzing and describing a system's behavior in terms of particular ways that the system is expected to be used" (Hsia et al. 1994) In the case of shared representation languages such as PSL, scenario analysis is defined as " the process of understanding, analyzing, and describing knowledge representation in the way the language is expected to be used.

One of the main objectives of this research study is the implementation of PSL in a construction scenario within the pre-construction process stages, to evaluate the use and the impact of this standard language in construction practices. This involves the representation of architectural design; pre-tender cost estimating; and scheduling and planning application areas related process information for a case study project and exchange of this information between these areas using PSL as a common standard language. In this implementation, PSL is expected to facilitate the exchange of correct semantics of the process information shared between the construction software applications chosen to support these areas; these are AutoCAD design, CCS cost estimating and Microsoft project scheduling and planning packages.

This section presents a description of a construction scenario, within the pre-construction process stages, chosen for the first trial implementation of PSL in the construction environment. This description includes; general overview of construction scenario; a brief overview of the generic construction process information representation and exchange within the pre-construction process stages, the architectural design, pre-tender cost estimating and scheduling processes, software applications supporting the design, cost estimating and scheduling processes; the on-site constriction processes that are normally described in the pre-construction process stages, and a typical construction example within the pre-construction process stages, which involves information representation and exchange using supporting software applications.

7.2 Implementation Overview

7.2.1 Purpose of the Implementation of PSL in a Construction Environment

The development of PSL has proceeded on an as-needed basis (Schlenoff, C , Gruninger M , Tissot, F , Valois, J , Lubell, J., Lee, J., 2000) and multiple pilot implementations of the language in manufacturing scenarios. The purpose (Raig Schlenoff, Mihai Ciocoiu, Don Libes, Michael Gruninger, 1999) of the multiple pilot implementations of the language each focusing on different package, within the manufacturing environment were to further develop and improve the initial process specification language, to ensure that it is able to handle real-world exchange scenarios, and to ensure that PSL can interface well with typical process-related software packages.

In the same way but in a different environment, the implementation of PSL in construction is aimed at exploring a construction scenario in which PSL can be used to facilitate the representation and exchange of process information, between candidate software applications of the scenario, for a the case study project. The purpose of this implementation of PSL in construction scenario is not to propose change to or extend the PSL ontology but to evaluate the use of the language in construction and the degree of applicability of the concepts of the language in providing common definitions for the construction process representations introduced in the scenario, and to identify the construction process representations that cannot be completely defined using the current PSL concepts and propose concepts and theories related to the scenario for new extensions development to the PSL ontology, and specify the relations that can exist between the existing process concepts of the PSL ontology and the new construction process representations semantic concepts.

7.2.2 Implementation Approach

The implementation of PSL in the construction scenario for a case study project involved the following steps:

1. Identification and description of a construction scenario for this research study and selection of software applications for the scenario.
2. Selection of a case study project for the construction scenario and analysis of the project construction specification to identity and gather process related information
3. Development of process models for the case study project construction and collection of data related to architectural design, pre-tender cost estimating, and scheduling stages for each activity of the process model
4. The scenario software applications' representations of the process related information identified and gathered from the project specification, and production of complete construction process information, for the scenario, using the software applications.
5. Identifying and clearly defining the construction process concepts of the scenario based on the data gathered from the case study project construction specification and the software applications' representations of these data
6. Exchange information between the scenario software applications using PSL
7. Identify and propose concepts and theories for related the scenario new extensions development to the PSL ontology.

7.3 Construction Context Overview

Generally, construction involves the implementation of the required facilities such as buildings, bridges, and roads etc. and the production of information, for the management of the implementation, at the various stages throughout the projects lifecycle. While, the implementation involves onsite construction operations (construction processes) carried out by contractors and subcontractors, the production of information for the management of the implementation spans from the conception of a project in a client's mind to design involving different disciplines (Architectural, Civil, Structural, Mechanical, Electrical, etc), cost estimating, construction process planning and scheduling, facility management etc In this research study, the implementation of PSL in a construction environment will focus on the exchanging of information between software applications supporting a construction scenario within the pre-construction process stages, which involve architectural design, pre-tender cost estimating, and scheduling, and planning.

Construction processes are complex with the subsequent need for management of a complex "construction process information" This information needs to be defined and managed prior to the commencement of the actual construction processes by contractors and the management of the construction is based on this information This information is generated and manipulated at different construction stages utilizing different software applications. Each of these applications focuses on particular aspects of a process. For insistence AutoCAD design applications supports the definition and specification of project products to be realised by the construction processes, the CCS cost estimating application generates resource and cost information of the processes, and Microsoft project planning and scheduling deals with the manipulation and management of time oriented process information. All this information is fragmented and needs to be shared across the different construction stages and between supporting software applications. While "Fragmented information" refers to the process information generated and represented at the various stages utilizing differing software applications with varying languages and terminologies; "Shared information" refers to the information shared, within the different construction stages and exchanged between supporting software applications used in construction stages. **Figure 7.1** outlines the construction process and process information representations and exchange scenario within the pre-construction process stages. This scenario involves the design, cost estimating, and scheduling, and planning information representations and exchange of the information between software applications supporting the scenario In this case while the implementation involves on-site construction operations the production of information for the management of the implementation involves the representation of the project products to be constructed, the activities performed to realize the products, the materials and resources required, the cost of the activities and their associated materials and resources, the construction plan; the time duration needed by the resources to perform the activities, and the relationships between activities, and constraints that may exist caused by resources availability, project products orientations and other controls. From **figure 7.1**, it can be seen that information is generated and represented by the different software applications during the pre-construction process stages, and this information is exchanged between the software applications supporting the design, cost estimating and scheduling process stages. The information generated by these applications is required during/for the implementation and management of the construction and sometimes after the commencement of construction, change of information is inevitable requiring the flow of information between construction processes, i.e. between onsite and offsite processes and between off site processes.

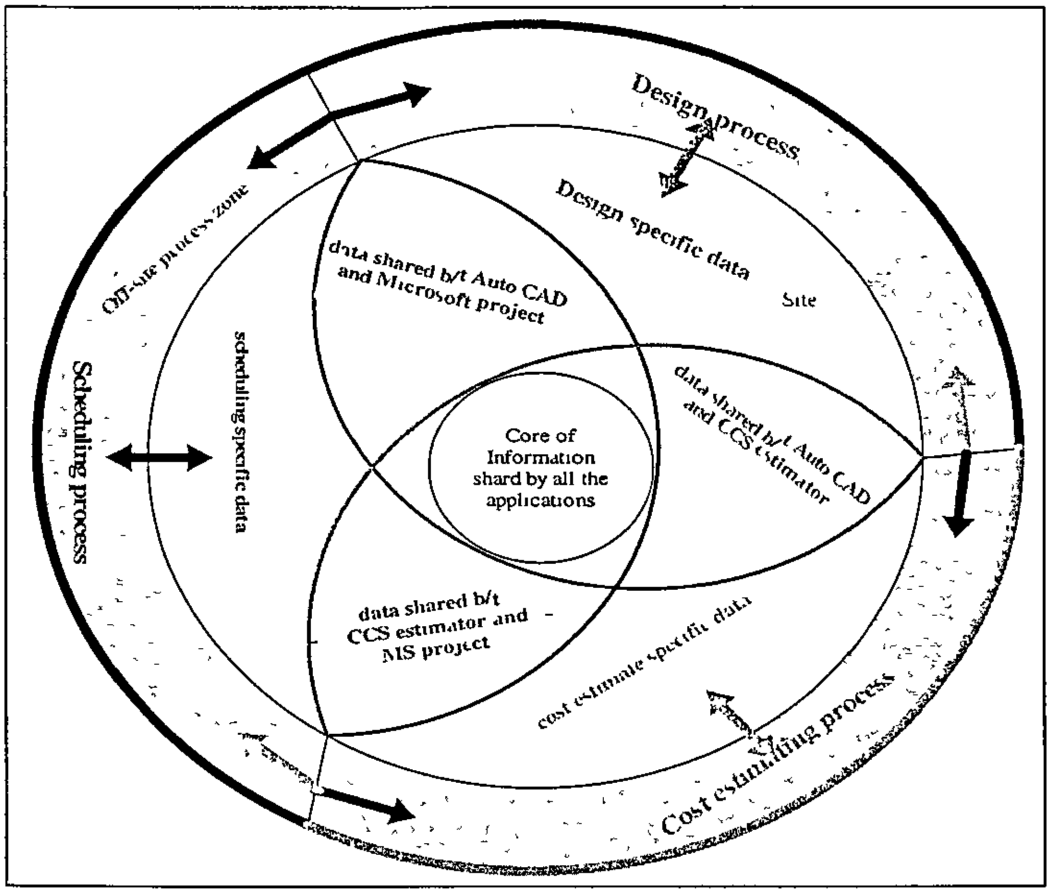

Design process
Design specific data
Site
data shared b/t Auto CAD and Microsoft project
Off-site process zone
Scheduling process
scheduling specific data
Core of Information shard by all the applications
data shared b/t Auto CAD and CCS estimator
data shared b/t CCS estimator and MS project
cost estimate specific data
Cost estimating process

Figure 7.1 Outline of Construction Scenario within the Pre-construction stages

7.4 Detailed Description of the Construction Scenario

This section presents a detailed description of the construction scenario within the pre-construction process stages. This provides a detailed view of the scenario, including process information representations within the scenario (architectural design, pre-tender cost estimating, and scheduling and planning processes), candidate software applications used in the scenario, an example of a construction situation, which involve real world type construction process information exchange, within the scenario, and an overview of project realization construction processes

7.4.1 Process Information Representations Overview

Process information representation refers to the sequence of activities carried out to describe processes needed to realise project products, including the high level activities, data requirements, resource requirements, ordering relations, time and durations requirements, cost, abstraction, location etc In the following sections, a detailed view of the archıtectural desıgn, pre-tender cost estımatıng, and scheduling and plannıng ınformatıon representations are presented and the supportıng software applıcatıons are descnbed

7.4.1.1 Architectural Design

Archıtectural desıgn is a process (an actıvity) of specıfyıng projects/facılıtıes/products such as buıldıngs and their parts, by means of drawings and textual specıficatıon, which complements the drawıngs This process (actıvıty) ınvolves several phases spannıng from bnef to schematic and detaıled design. Schematıc design is the phase where the gross features of the product or facılıty are defined includıng the overall shape, size, structure, adjacencıes, cırculatıon, and the matenals of construction, and each feature wıll undergo refinement and elaboratıon in subsequent desıgn phases, the detaıled desıgn. Based on the VTT buıldıng process model (V , Keıtıla, M , Lahdenpera, P. 1997) a formal representatıon of an archıtectural desıgn process was developed usıng an IDEF3 Process modelıng Methodology (Mayer et. Al 1992). Both, the process and object vıew of the model ıs attached ın **appendix B. Figure 7.2** shows the process vıew of the decomposıtıon of the "produce and manage archıtectural design data" process, which ıs the sub process of the overall constructıon process.

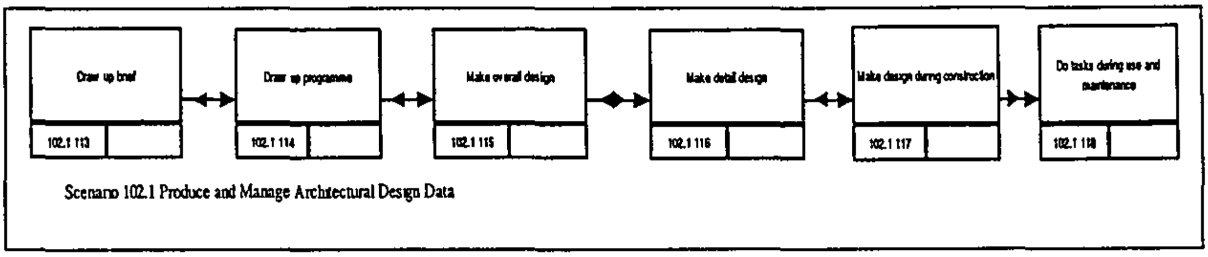

Draw up brief
102.1 113
Draw up programme
102.1 114
Make overall design
102.1 115
Make detail design
102.1 116
Make design during construction
102.1 117
Do tasks during use and maintenance
102.1 118
Scenario 102.1 Produce and Manage Architectural Design Data

Fıgure 7.2. IDEF3 model for the produce and manage archıtectural desıgn data hıgh level process

Takıng the view that projects are unıque assemblies of dıscrete, mostly standardızed components, the design and representation of projects ınformatıon is dıvıded ınto two components: the production of project object, using drawıng, which stores ınformatıon, about vanous project elements usıng geometncal representations; and the production of specıficatıon, text files, which holds textual ınformation about these elements and relatıonshıps that can exıst between them. Drawıngs represent geometry of objects, in a hıghly symbolic form accompanied by textual labels and specıficatıons (such as materials, products, construction techniques, etc). The representations must include the necessary information that can support the assessment of the project from different points of view such as production, cost estimating and scheduling, and planning are among many others. The results of this process are. hierarchy of products (products and breakdown of these products) to be constructed by hierarchically interconnected construction processes.

7.4.1.2 Pre-Tender Cost Estimating

The intent of cost estimating is to determine the monetary value of projects based on various objects, involved in the project realization processes and operations, including resources used and consumed A cost estimator's task involves several sub-processes to determine the estimated cost of an activity. Determination of the purpose and scope of the estimate is the first step and the identification of the activities, products and resource to perform the activities follow. The total sum of the costs of all the activities gives the cost estimate of the project Cost of an activity required to construct or install a designed object/product is estimated based on the use and consumptions of resources performing the activity. This requires design information and product data of projects i e. detailed information of the designed project object, its component parts, material of construction, method of construction, dimensions, quantities etc Based on this information, tasks or activities and materials are selected and resources allocated resulting in the cost of the activity based on the cost rate of the resources and materials. This estimate includes the unit costs of resources, the resulting costs of the tasks, and the total costs of the project

Based on the Data Flow Diagrams (DFD) cost estimating procedure (Norman 1992) a formal representation of the cost estimating process is developed using an IDEF3 Process Modeling Methodology (Mayer et. Al 1992). Both, the process and object view of the model is attached in **appendix B. Figure 7.3** shows the process view of the decomposition of the "pre-tender procedure", which is the sub process of the overall construction operation.

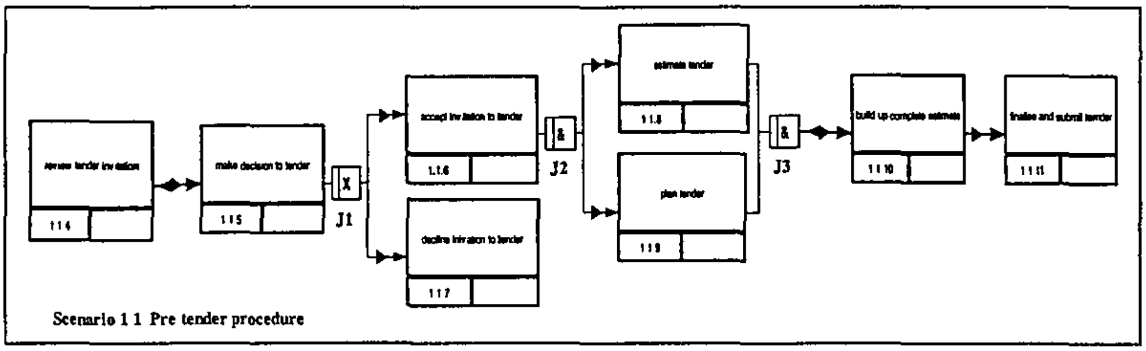

review tender invitation
1 1 4
make decision to tender
1 1 5
X
J1
accept invitation to tender
1.1.6
decline invitation to tender
1 1 7
&
J2
estimate tender
1 1 8
plan tender
1 1 9
&
J3
build up complete estimate
1 1 10
finalise and submit tender
1 1 11
Scenario 1 1 Pre tender procedure

Figure 7 3 IDEF3 model for the pre-tender procedure: cost estimating high level process

Cost Estimating process includes the following Sequence of Activities:

- Analysis of design data available for cost estimating taking in to consideration the purpose and scope of the estimate and determination of the material of construction and construction method The construction method specifies the tasks that need to be completed to construct the design objects and the material of construction is the material consumed by a task.
- Identification of resources that may be needed to perform the tasks realizing/constructing the designed objects, the resources required to perform each of the tasks are determined according to their performance and types
- Once the types and quantities of resources are determined for an activity, costs are applied to resources and cost of a task is calculated.
- Cost estimate can be carried in different forms, cost based on tasks and resources suing unit costs for the resources or cost based on the overall product using unit cost for the overall object.
- Cost may be summarized at the task (for process management data) or product (for design object) level.

7.4.1.3 Scheduling and Planning

Projects are made up of tasks and each task has work to be done and the completion of these tasks is important to the project's completion. Scheduling is an activity of planning and organizing tasks, estimating time and duration and sequencing of work required to construct a project object/building based on resource usage and relationships between the tasks and specification of the effects of these on the total time-duration of the project to be delivered. A task in a schedule is a job that has a beginning and an ending and it sometimes referred to as an activity Scheduling requires design and/or cost estimating information and breakdown of complex work or aggregation of simple estimating data into tasks appropriate for scheduling

Schedules consist mainly of tasks, dependencies between the tasks, durations, constraints, and time-oriented project information. Different scheduling terms are used to represent scheduling information. For instance, task dependency constrains start and Finnish time, Lead-time overlap tasks; Lag time delay tasks; tasks may split for temporary task halt e g. during redesign of part of a project its production activity should be on temporary halt, a task is a predecessor or successor, repetitive, or critical in a schedule, precedence constraints define the ordering relations between activities. Tasks in a schedule require resources, which may be recoverable or non-recoverable during the activity A resource becomes non-recoverable if a task requires that it will not be available to perform other tasks in a schedule and a resource becomes recoverable if a task requires a resource that will be available for other tasks in a schedule.

Based on the Data Flow Diagrams (DFD) cost estimating procedure (Norman 1992) a formal representation of the scheduling and planning process is developed using an IDEF3 Process Modeling Methodology (Mayer et. Al 1992). Both, the process and object view of the model is attached in **appendix B**. The sequence of activities involved in the scheduling and planning process are formally represented in this model **Figure 7.4** shows the process view of the decomposition of the "plan and schedule construction operations" process, which is the sub process of the overall construction operation.

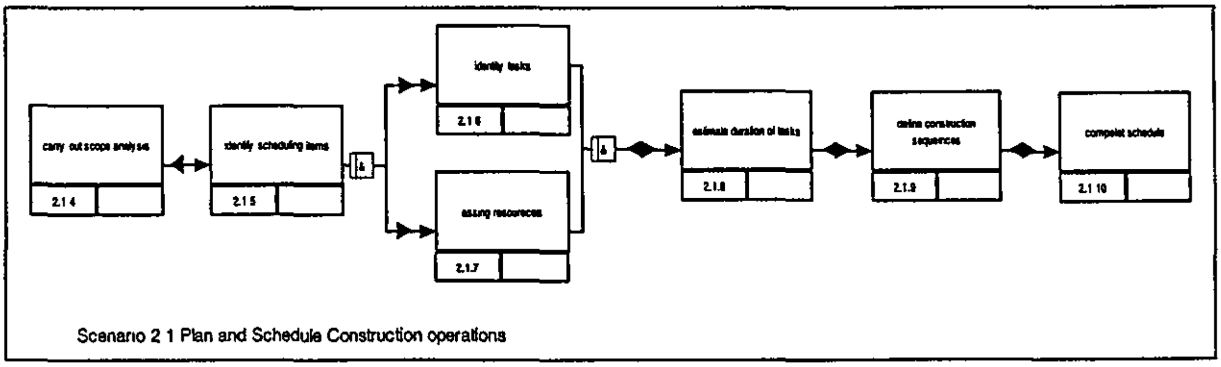

carry out scope analysis
2.1 4
identify scheduling items
2.1 5
&
identify tasks
2.1 6
assing resources
2.1.7
&
estimate duration of tasks
2.1.8
define construction sequences
2.1.9
compelet schedule
2.1 10
Scenario 2 1 Plan and Schedule Construction operations

Figure 7.4. IDEF3 model for the panning & scheduling high level process

7.4.2 Software Applications Supporting the Process Information Representations

There are numerous computer applications currently used to support the representation and management of process information in the pre-construction process stages. AutoCAD, CCS and Microsoft project among others are the candidate software applications of the scenario selected to support process related architectural design, pre-tender cost estimating, and scheduling and planning information representations. Technical and use descriptions of the candidate software applications will be further elaborated in the following sections. This includes how these applications support the representation of information in projects, the information representation and exchange formats and input and output data of each of the applications

7.4.2.1 AutoCAD

AutoCAD is a general-purpose computer aided drafting tool, initially developed in the early 1980's by AutoDesk Inc. It is an AutoLISP program for creating and manipulating AutoCAD graphical and textural objects The tool supports several programming languages, from simple scripts to a C++ interface. This application consists of AutoLISP applications or routines that provide drafting objects such as text and symbol fonts, line types, shapes, hatch patterns, fill patterns, and geometric characteristic symbols. These objects enable users to create AutoCAD drawing objects such as buildings and their parts, and attach attributes for textual specification, which characterises the drawing objects.

At the core of any computational system that can support design development, there is an intelligent representation, which should be able to represent all the different components that make up a project, along with the manner in which they come together The representation must be geometrically clear and textually informative complementing geometrical representation. AutoCAD uses graphical drawings (DWG files) and attributes (ASCII text-files) to represent project product data for project products realisation and other purposes.

The drawing objects provide a geometrical representation of a project such as a building mass (during the scheme design stage), compound object parts of a project such as doors and windows and their detailed component parts (such as frame, panel, hardware set etc.)

in the detailed design stage. The detailed geometrical representation of a project provides information such as dimensional data, object's location (to the required level of detail). AutoCAD includes tools that are used to create and edit the representations, drawings and textual attributes. Block-objects provide the capabilities of organising and manipulating components into compound objects and this may be exploded in order to reduce it back to its separate components for detailed information Attributes can be used to attach information to blocks or drawings to convey important information that can later be extracted from a drawing for use in a spreadsheet or database to produce information such as parts list or bill of materials (BOM) and bill of quantities (BoQ) **Tables 7.1 and 7.2** illustrate AutoCAD Design output data types and the file formats respectively.

AutoCAD Design Software
Output Data types
Drawings: - Graphical/geometrical information representing of the design object and its dimensional values Such as distance, angles etc. **Textual information such as:** - Dimensional values of a drawing representing the designed object. This can be area, volume etc. - Material of the designed object - Method of construction/production
These data can be represented in the following AutoCAD native file formats:
Drawings. graphical/geometrical representation **Blocks:** symbols representing a drawings **Attribute (ASCII text file):** a tag attached to a block containing data for a drawing inserted in to a block. Such data can be. - Dimensional values - Material - Method of construction/production **External references:** - store drawings as external reference. Similar to blocks **Drawing databases.** - for textual Information extracted from blocks, External references etc. **Database files·** - Attribute information extracted from DWGs formatted in a spreadsheet or database files.

Table 7.1 AutoCAD Design output data representations

AutoCAD Output (export) file formats
DXF (a dos text file)
DXF is ASCII or Binary file
Text
Spread sheet
Database
Metafile (wmt)
ACIS(sat)
Litograph (st)
Encapsulated (eps)
DXX extract(dxx)
Bitmap graphics (bmp)
AutoCAD other releases (dxf)
Raster and vector files
Postscript description
HTML format
DWF (drawing web format) files
DXB files(Dumb graphics)

Table 7 2. AutoCAD Design file formats

In addition to the native file formats (DWG and textural attributes) AutoCAD can read from and write to the external file formats shown in **figure 7.3.**

the complete list that appears in the AutoCAD File import and export and File save dialog drop-down		
import file types:	**export file types:**	**save file types:**
.wmf (Windows Metafile) **dxf** (Drawing interchange file) **.sat** (ACIS files) **esp** (Encapsulated PostScript files) **pcx** (raster-image file) **.tif** (raster-image file) **.gif** (Bitmapped picture format) 3ds (3D Studio files)	**Wmf** (Windows Metafile) **.esp** (Encapsulated PostScript files) **.sat** ACIS files) **.dxx** (DXX extract) **.bmp** (bitmap graphics) dxf **(a dos text file, (ASCII or Binary file))**	SAVEIMG saves a rendered image to a file in TGA, TIFF, GIF, or RND format. BMP File It saves selected objects to a file in device-independent bitmap format

Table 7.3 AutoCAD application import /export and file save types/formats

7.4.2.2 Construction Computer Software (CCS)

Construction Computer Software (CCS) is an integrated Project Management System specifically designed for the construction industry (WWW.ccsuk.com). The major components are Estimating, Valuations, Project Planning and Project Cash Flow, used by project planners and contractors for greater accuracy and productivity. CCS was started in 1978 as a division of a large South African construction company. Today it is an independent organization and most of the software team has been with CCS since its inception. Over 250 contractors in more than 20 countries use the CCS System. The system is continually being extended to accommodate changing conditions in the industry and new user requirements

The CCS estimating application uses ASCII text files to specify the tasks and resources, details of the project and materials, equipment and labour and other related information. **Table 7.4** illustrates the input and output data types of the CCS applications. The file input and output format are shown in **table 7.5**. The following bill importers are available in CCS to facilitate the import of information form other applications supporting the; Delimited files (CSV), ASCII files (Text), CITE files and WinQS files.

CCS Cost Estimating INPUT/OUTPUT DATA TYPES	
INPUT DATA	*OUT PUT DATA*
BILL OF QUANTITIES □ PAGE □ ITEM □ MACRO OP-CODE □ MACRO OP DESCRIPTION (BILL ITEM DESCRIPTION) □ UNIT □ QUANTITY of bill item □ NETT RATE □ NETT AMOUNT	BILL OF QUANTITIES GROSS AMOUNT ITEM OP-CODE DESCRIPTION (BILL ITEM DESCRIPTION) UNIT QUANTITY of bill item GROSS AMOUNT (gross-rate *bill-qty)/sum of split SPLIT AMOUNTS □ LABOUR --(nett-split rate *bill-qty) □ PLANT--(nett-split rate * bill-qty) □ MATERIAL--(nett-split rate * bill-qty) □ TEMP. MATERIAL--(nett-split rate * bill-qty) □ OVERHEADS--(nett-split rate * bill-qty) □ PROV SUMS--(nett-split rate * bill-qty) □ SUBCONTRACT--(nett-split rate * bill-qty) □ MARKUP
MACRO OP □ ITEM □ OP CODE □ OP CODE DESCRIPTION (BILL ITME DESCRIPTION) □ UNIT □ QUANTITY □ NETT RATE □ NETT AMNOUNT	*BILL ITEM INFORMATION* ITEM CODE DESCRIPTION QTY UNIT NETT-RATE =(SUM OF RESOURCE RATES) **RESOURCES INFORMATION** □ RESOURCE GROUP □ RESOURCE CODE □ RESOURCE NAME □ CURRANCY □ QUANTITY □ RATE □ AMOUNT =QTY*RATE □ PRODUCTION
SPLIT RATE PRICING BILL ORDER □ OP CODE □ PAGE/ITEM □ MATERIAL □ LABOUR □ PALNT □ TEMP. MATERIAL □ OVERHEADS □ PROV SUM □ SUBCONTRACT □ NETT RATE □ NETT AMOUNT	

Table 7 4 Construction Computer Software (CCS) application input output data types

In addition to the native file format CCS can read from and write to the following external file formats.

CCS can writes to **MPX files** that may be opened in Microsoft Project or Primavera P3, writes to and reads from Microsoft Excel **files**, writes to multiple Excel spreadsheets **files** for use with other applications, reads from a Microsoft Project export file (**MPX file**), reads from a Primavera export file, reads from an ASCII (text) file, reads **Lotus 1-2-3** through Microsoft Excel, reads from Microsoft word and Save current selection as Bitmap file.

Input File formats	*Output File formats*
Space delimited text	Spool file (direct)
Tab delimited text	PDF export
Comma delimited text	ASCII (Text)
Comma-Quote delimited	Comma separated (CSV)
Quote-Quote delimited	Excel Spread Sheet (direct)

Table 7 5 CCS Input/Output File formats

7.4.2.3 Microsoft Project Scheduler

Microsoft Project is a scheduling tool developed by Microsoft inc., currently widely used in the construction project management domain Microsoft project scheduler is a C++ library of classes and functions that implement concepts for constraint based scheduling. The library enables the representation of scheduling elements and constraints such as activity duration, precedence constraints, resource assignments and resource sharing. The scheduler uses MPP, MPX and templates to specify and represent project information

MS project uses DFD cost estimating procedure to assign cost rates to resource's work i e. work items or BoQs or activities are assigned with cost rates that include elements for the people, equipment, and supplies used to complete tasks in a project. Microsoft Project also allows assignment of rates to resources so project costs can be managed accurately. A cost that remains constant regardless of the task duration or the work performed by a resource can be assigned to a task. Multiple standard rates, overtime rates, or per-use rates can be assigned to resources along with the dates for each rate to go into effect. MS Project also allows the comparison of current cost, or budget This application requires design and/or cost estimating data as input and produces the construction process plan and schedule. The Input-output data file of MS Project application is shown in **table 7.6.**

Microsoft project also introduces the notion of groups and units of resources required by an activity Resources in Microsoft project include the people, equipment, supplies, subcontractors, and preliminaries. The following cost notions: fixed task cost, data flow diagram (DFD) cost estimating rates for tasks and multiple standard rates such as over time rates and per-use rates to resources are used in the application Microsoft project scheduler defines precedence constraints and time-variations concepts The precedence constraints are finish-to-start, start-to-start, finish-to-finish, and start-to-finish The time-variations are lag-time, lead-time and various slack-times. Activities also consume and use resources to realize the project products Microsoft project supports schedule information management, using scheduling applications representations such as Gantt and PERT charts.

MICORSOFT PROUJECT SCHEDULING	INPUT OUTPUT DATA TYPES
INPUT DATA	**OUT PUT DATA**
PROJECT □ START AND FINISH TIME	PROJECT □ START AND FINISH TIME
TASK INFORMATION □ ID □ WBS □ CONSTRAINTS □ TASK NAME □ DURATION □ START TIME □ FINISH TIME □ PREDECESSORS □ RESOURCE NAMES	TASK INFORMATION □ ID □ WBS □ CONSTRAINTS □ TASK NAME □ DURATION □ START TIME □ FINISH TIME □ PREDECESSORS □ RESOURCE NAMES
RESOURCE INFORMATION □ ID □ RESOURCE INFORMATION □ RESOURCE NAMES □ INITIAL □ GROUP □ MAX UNIT □ STD RATE □ OVT RATE □ COST/USE □ ACCRUE AT □ BASE CALENDAR □ CODE	RESOURCE INFORMATION □ ID □ RESOURCE INFORMATION □ RESOURCE NAMES □ INITIAL □ GROUP □ MAX UNIT □ STD RATE □ OVT RATE □ COST/USE □ ACCRUE AT □ BASE CALENDAR □ CODE
TASK FIXED COST INFORMATION □ ID □ TASK NAME □ FIXED COST □ FIXED COST ACCURAL □ TOTAL COST □ BASE LINE □ VARIANCE □ ACTUAL □ REMAINING	TASK FIXED COST INFORMATION □ ID □ TASK NAME □ FIXED COST □ FIXED COST ACCURAL □ TOTAL COST □ BASE LINE □ VARIANCE □ ACTUAL □ REMAINING

Table 7 6 Input-output data types for Microsoft project scheduling applications

Table 7.7 shows the complete list that appears in the Microsoft Project **File Save and File Open** dialog file-type drop-down. Microsoft Project 98 can read from and write to these external file formats.

File Save file types	*File Open file types*
Project (* mpp)	All Files (* *)
Template (* mpt)	Microsoft Project Files (* mp*)
Project Database (* mpd)	Projects (* mpp)
MPX 4 0 (* mpx)	Project Databases (* mpd)
Microsoft Access 97 Database (* mdb)	Templates (* mpt)
Microsoft Excel Workbook (* xls)	Workspaces (* mpw)
Microsoft Excel PivotTable (* xls)	MPX (* mpx)
HTML Document (* html)	Microsoft Access Databases (* mdb)
Text (Tab delimited) (* txt)	Microsoft Excel Workbooks (* xls)
CSV (Comma delimited) (* csv)	Text (* txt)
	CSV (* csv)

Table 7 7 Microsoft project scheduling file save and file open formats

7.4.3 Construction Projects Realization Process Overview

In the description of the construction scenario and throughout the report the term " process information" has been the keyword for this research study and the definition of this terms is given in the report. However the actual "construction process" which is the basis for this key word has not been described. Hence this section provides description of the construction process in terms of construction projects plan or schedule, cost estimate and design

Construction process refers to the operations that correspond to sets of activities or tasks, which can be performed to construct or produce numerous separate parts of a project and eventually achieve the final required facility such as a building. Each of the activities or tasks is the result of decisions concerning the method of construction to be used in order to achieve the goal, a constructed facility. The process of decomposing this high level process to low-level detailed activities, which are executed during production process, results in hierarchal process network (hierarchically connected operations realizing the several parts of the facility and eventually achieving the final facility). A construction project schedule or plan represents construction processes as a collection of interrelated tasks along with other scheduling information such as time duration, resources, relationships between the activities etc. In the same way, a cost estimate of a construction project represents construction processes as cost estimating tasks along with other cost estimating information such as resources performing tasks and cost of the resources. Also a design of construction project represents construction processes a set of construction methods along with other design data such as material of construction, locations, products of the design methods, and so on.

7.4.4 An Example of a Construction Situation within the Scenario

In a design and build project contract, a designer is required to produce a design for a project to a specified budget and time duration for construction to commence immediately after approval is obtained for the design and cost estimate from the client among others. The project involved the architect designer, the cost estimator, and the scheduler or planner The designer's role is to produce a design data The cost estimator, estimates the cost of the activities based on the cost of the materials and resources, the planner takes design and/or cost estimating data, aggregates or identifies scheduling tasks with single deliverables, that are necessary to produce the products within the project, assigns resources required by the activities, calculates the time and duration required, and determines the order in which these activities must occur taking into account any constraints etc.

In this situation, since architectural design is a multi-phase process, at the various stages during the evolution of the design, an estimator is required to produce the cost estimate in order for the designer to keep to the specified cost. During this process the estimator requires design data and the designer need to access cost data for the design objects produced. A planner is required to produce an acceptable construction process plan based on the design and/or cost estimating data, for the construction process to start after the design and cost estimate approval. For instance, during the early stages very little design information will be available and only a brief estimate, based on the information available will be possible In later design stages, when more detailed design information is available, a more accurate estimate can be produced. At the same time, during the design and estimating processes, since design is a top-down and production is a bottom-up operation, only an interim plan of the construction operation can be produced. This includes planning tasks, assignment of resources and cost, and estimate of time-duration of each task. For the construction process to commence immediately after approval is obtained for the design and cost estimate the design, cost estimating and construction scheduling need to be more or less concurrent. The challenge is that the designer uses the AutoCAD, the cost estimator uses CCS cost estimating and the scheduler uses Microsoft proȷect, whıle as descnbed above each of these applications uses dıfferent ınformatıon representatıons, data formats and termınology These applıcatıons currently cannot exchange ınformation.

In thıs research study ıt was decıded to use PSL, ın such a sıtuatıon wıthın the scenarıo, as a neutral, standard language to serve as an ınterlingua to ıntegrate and allow these applıcatıons to share semantıc concepts of theır respectıve ınformatıon representatıons Hence, the desıgner produces a desıgn data file usıng AutoCAD as shown ın the IDEF3 model of the desıgn process (see **Appendix B**) and runs the translator to convert the file to PSL The cost estımator and scheduler then run theır respectıve translators to convert the desıgn file ın PSL to CCS and Microsoft proȷect respectıvely and produce the requıred cost estımate and process plan, followıng the tasks/procedures of the cost estımatıng and plannıng IDEF3 model (see **Appendix B**) The use of PSL ın the process ınformation exchange between the constructıon scenarıo software applıcatıons ıs shown through the implementatıon of the language in the constructıon scenarıo wıthın the pre-constructıon process stages, descnbed.

Chapter 8 Implementation of PSL in the Construction Scenario

8.1 Introduction

In chapter 7, a construction scenario within the pre-construction process stages was detailed and an example of a construction situation was described within the scenario. This chapter presents the implementation of PSL in the construction scenario. The implementation of PSL in this scenario required a construction case study project, and a "metal door in a metal opening of a building project" was selected in order to have a real construction process information exchange, between the scenario software applications, using PSL as interchange language in the trial of the first implementation of PSL in the construction environment.

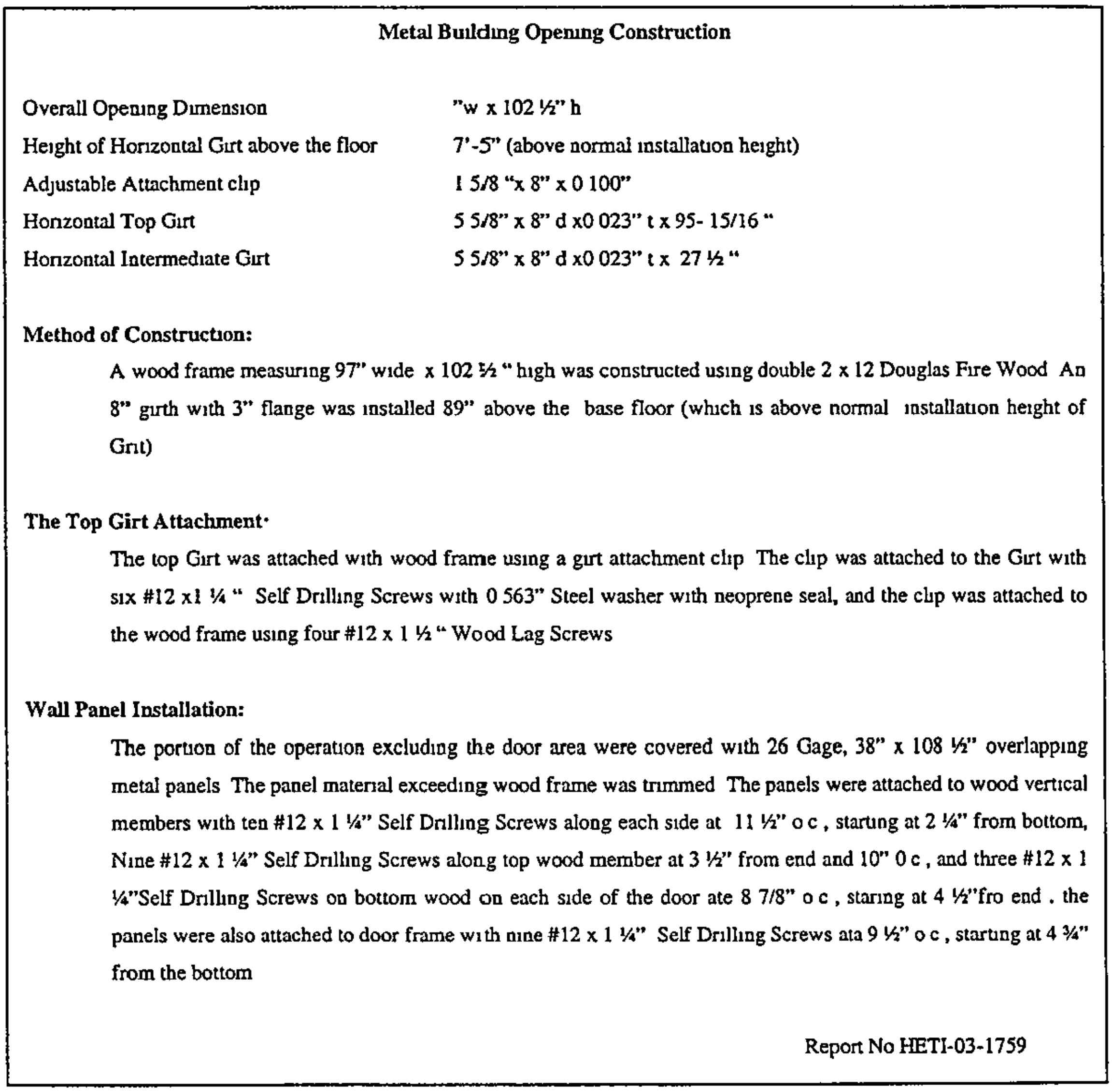
This section presents a detailed description of the construction case study project, "a metal door in a metal opening of a building project". The PSL language was implemented in the scenario in which the language can be used as an interchange language for the case study project construction process related architectural design; pre-tender cost estimating, and scheduling information exchange between the scenario software applications. For example, as described in the example of a construction situation within the scenario in the pervious chapter, the result of a design task need to be exchanged between the cost estimator and scheduler applications to cost estimate and plan the construction processes. However the three applications may have different interpretations for their respective construction process representations. The correct interpretation and exchange of this information can be achieved through the use of PSL as a common interchange language between the applications. However, prior to the use of the language in the construction environment the need to evaluate the use and impact of the language in construction practices became important. As a result this project determined to implement the language in the construction scenario described in which process related architectural design, pre-tender cost estimating, and planning process information, for construction of a metal door in a metal opening of a building project, can be exchanged between AutoCAD, CCS and Microsoft project applications using PSL. A manual translation process was carried out to exchange, the case study project construction process information, between the scenario software applications using PSL based hand written translators. In the following sections a detailed description of the case study project is presented, followed by the result of the translation process.

In the following sections a detailed description of the case study project "the metal door in a metal opening of a building project" within the scenario is presented. **Section 8.2** presents descriptions of the design products to be constructed or realised (construction products) and the materials of construction used for these products (raw materials or manufactured products), and **section 8.3** describes the processes that may be used to assemble or construct the construction products. **Section 8.4** describes the structuring of the construction process related information of the case study project using the scenario software applications' information representations, file formats, and terminologies. In this section the architectural design data is presented and structured in Excel file format, the cost estimating data is presented using the CCS application's data representation format, and the construction planning information is structured using the Microsoft project scheduling information representation format **Section 8.5** describes the manual translation process, by which the cases study project construction process information was exchanged between the scenario software applications using PSL and the applications based hand written translators Finally a section of discussion on the implementation of PSL in the scenario is presented and conclusions and recommendations are drawn.

8.2 Cases Study: "Metal Door in a Metal Opening of a building" Project

This section presents a detailed description of a situation in which PSL may be used as a common representation for architectural design, pre-tender cost estimating, and scheduling and planning related process information for the case study scenario project This representation is expected to facilitate the exchange of this process information between the scenario software applications. A textual specification of the construction case study project, which includes "metal building opening construction detail" and "metal door construction detail" is shown in **figures 8.1 and 8.2,** and detailed drawings of the building opening and the door parts are attached in **appendix E.**

These types of doors are mostly used for building in hurricane vulnerable areas and their construction detail is complex This type of project with complex construction information has been selected for the construction scenario to illustrate the complex process related information in construction projects that need to be represented unambiguously, shared and exchanged between applications consistently The source of the design specification of the case study project comes from Hurricane Engineering and Testing INC.https://secure.metaldoor net/dominionproducts/specifications/specs110 cfm, which deals with computer controlled product testing and design and wind analysis of construction products. Sets of information were abstracted from the textual specification of the construction case study project and described in the following sections. These descriptions provide an overview of the various elements, which represent the case study project. These are the products, materials (resource) and construction processes aspects of the case study project These aspects came from the textual specification of the case study project hence relate to the design stage of the construction scenario and referred to as the process information related to design. Other process information related to cost estimating and scheduling or planning were identified based on the design related process information of the project and structured using the CCS and Microsoft project scheduling applications' information representations described and shown in **section 8.4**



Metal Building Opening Construction

Overall Opening Dimension	"w x 102 ½" h
Height of Horizontal Girt above the floor	7'-5" (above normal installation height)
Adjustable Attachment clip	1 5/8 "x 8" x 0 100"
Horizontal Top Girt	5 5/8" x 8" d x0 023" t x 95- 15/16 "
Horizontal Intermediate Girt	5 5/8" x 8" d x0 023" t x 27 ½ "

Method of Construction:

A wood frame measuring 97" wide x 102 ½ " high was constructed using double 2 x 12 Douglas Fire Wood An 8" girth with 3" flange was installed 89" above the base floor (which is above normal installation height of Grit)

The Top Girt Attachment·

The top Girt was attached with wood frame using a girt attachment clip The clip was attached to the Girt with six #12 x1 ¼ " Self Drilling Screws with 0 563" Steel washer with neoprene seal, and the clip was attached to the wood frame using four #12 x 1 ½ " Wood Lag Screws

Wall Panel Installation:

The portion of the operation excluding the door area were covered with 26 Gage, 38" x 108 ½" overlapping metal panels The panel material exceeding wood frame was trimmed The panels were attached to wood vertical members with ten #12 x 1 ¼" Self Drilling Screws along each side at 11 ½" o c , starting at 2 ¼" from bottom, Nine #12 x 1 ¼" Self Drilling Screws along top wood member at 3 ½" from end and 10" 0 c , and three #12 x 1 ¼"Self Drilling Screws on bottom wood on each side of the door ate 8 7/8" o c , staring at 4 ½"fro end . the panels were also attached to door frame with nine #12 x 1 ¼" Self Drilling Screws ata 9 ½" o c , starting at 4 ¾" from the bottom

Report No HETI-03-1759

Table 8 1. Metal Building Opening Construction Design Detail (source· Hurricane Engineering and Testing INC)

Door Construction Details

PRODUCT	Single Metal Door Installed in Metal Building Opening

DESCRIPTION OF UNIT

Model Designation	AMSCOKD, 20 Gage Textured
Overall door size	41" x86 1/8" h
Configuration	X
No and size of vents	(1) 35 ¾" x 83 3/8" (active)

8.1 Door Frame Material & Construction

Strut	A sliding 1 3/4"x8" galvanized steel strut with 2 ½"x3 7/8x wide welded steel clip on top The assembly was slid upward to attach to horizontal girt using two 5/6" X ¾" nut and bolt on interior and exterior face of jamb
Jamb and Head	16 Gage Kerfed frame profile, double rabbet with a foam filled Gasket, measuring 3"X81/8" (overall dimension) The depth of the door stop was 2 1/8" in front and 1 15/16" in rear, the height of door stop was 5/8"
Hinge reinforcement	three, 7 Gage, 10" x 1 ¼"" hinge reinforcement were four point spot welded at end to doorjamb at 7" from end at centre of the jamb
Bottom Jamb angle	12 Gage galvanized steel angle measuring 1 ¾" x2" x7 ½" x 0 099"thick 1 ¾" leg was four point spot welded to the jamb
Strike Plate	4 86" x 1 75" x0 079" thick steel strike plate was installed to the doorjamb
Strike Plate Reinforcement	A three point spot welded 16 gage x 1 ¼" x6" strike plate reinforcement was installed
Corner Construction	The door jamb was attached to head extension strips with two 5/16" x ¾" hex head nut and bolt screws

8.2.1
Door Leaf Material and Construction

Overall size	35 ¾" wide x 83 1/8" high x 1 3/4 " deep
Core	Expanded polystyrene core
Hinge Reinforcement	(3) 9" x 1 ¼" x 7 Gage
Lock Preparation	A 16 Gage steel lock reinforcement plate was six points spots welded to the door face sheet.
Leaf Construction	20 Gage galvanized steel top and bottom face sheet with vertical hemmed edge seams, mechanically interlocked, square hinge and 1/8" in 2" bevelled lock edges A 16 Gage flush top and bottom channel welded to both face sheets. Hinge reinforcement was installed at centre of the leaf and 6 5/16" from top and bottom

Hardware and Components

Locks	Yale 5307 Lever Lock (one)
Hinge	(3) 41/2" steel hinges with non-removable pin template
Weatherstrip	AMSCO extruded aluminium & vinyl weatherstrip.
Method of attachment	The door assembly was attached to horizontal grit on top and wood frame ate bottom using adjustable clips Jamb mounted to girt using ¼*1" Machine screws 2 each side, and to wood frame using 1/16"*1 5/8" wood lag 2 each side

Table 8.2 Metal Building Opening Construction Design Detail (source. Hurricane Engineering and Testing INC ,)

8.2.1 Products and Materials Overview of the Case Study project

The "metal door in a metal building opening" is a product of a sub-process within the construction of a building, which can be performed to assemble the component parts of the product. Various resources are used throughout the construction processes of the metal door and these may be used in different forms during each process These are referenced in the detailed description of the construction processes A diagrammatical view of the design products of the case study project, and the materials, which may be manufactured products or raw materials that are required during the construction process of the project are depicted in **figure 8.1.** This diagram shows the hierarchical breakdown of the products and materials of the case study project

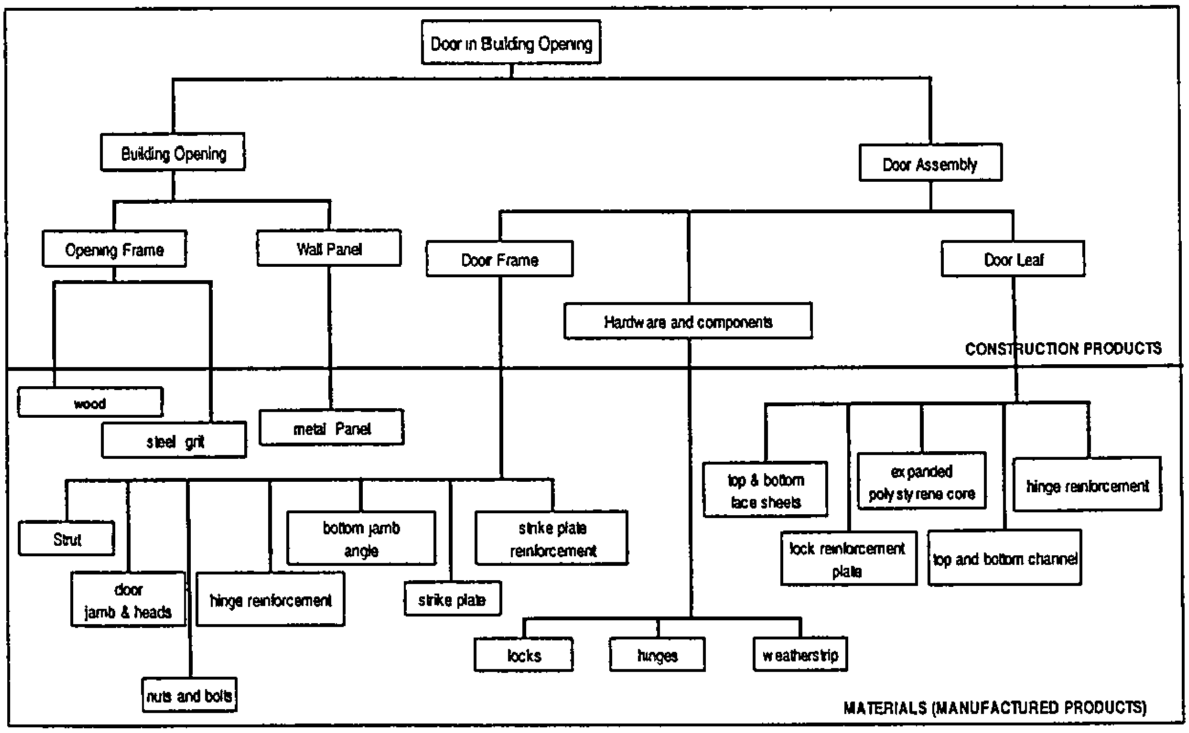


Figure 8.1 . hierarchical breakdowns of the products and materials of the case study project

8.3 Detailed View of the Case Study Project Construction Processes

In this section, a particular set of interrelated processes and sub-processes, and the activities, which represent the decomposition of these processes and sub-processes, for construction of the case study project are described. These processes and activities can be used in the generation of a detailed process plan and cost estimate of the construction of the "metal door in a metal opening of building project" **Figure 8.2** Outlines a generic high level construction processes for a typical building project and illustrates the relationships between the various processes. The relationships between the processes can be read from top to bottom with the most general at the top and detailed at the bottom. A more detailed view of the two sub-processes for the "building opening" and "complimentary structure door" construction highlighted in **figure 8.2**, which constitute the "metal door in a metal opening of a building project" construction process, are shown in **figure 8.3**

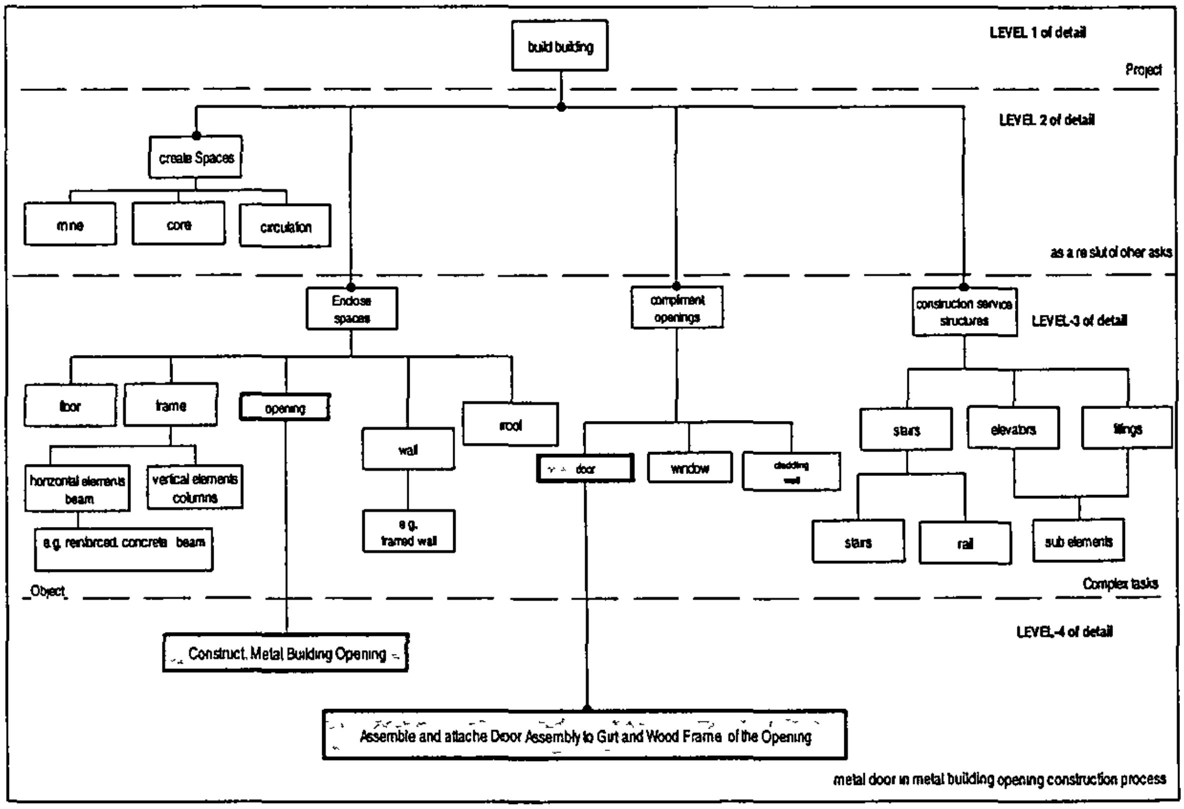


Figure 8 2. Generic Overview of Building Construction Processes

The following sections (sections 8 3 1 to 8 3.9) include detailed descriptions for the "metal door in metal opening of a building project" construction processes shown in **figure 8.3.** This includes descriptions for the sub-processes and activities that must be completed for the accomplishment of the high level "construct metal door in metal opening of a building project" process. The processes, sub-processes and activities are extracted from the construction specification of the cases study project shown in tables

8.1 & 8 2. The detailed description of each of these processes, sub processes and activities contains an IDEF3 Process Description Capture method representation (IDEF3 process model) and textual specifications for each of the activities of the process model. The processes and sub-processes mainly reflect the hierarchically interconnected activities, which are used to achieve the parts of the "metal door in a metal opening of a building project". At the lowest level of decomposition of each process i e. at the activity levels, the various resources required and products produced that are data related to design stage of the scenario are listed. Process data related to the cost estimating and scheduling stages can also be attached to each activity of the process model, but due to the dimension of data that need to be listed for each activity, and to keep this clear and uncluttered these data are presented using the scenario software applications Based on the design related data for each activity of the process model other process information related to cost estimating and scheduling for each activity of the process model are identified and organized and structured using the scenario software applications' information representation files and screen shots of these representations are shown in **section 8.4**

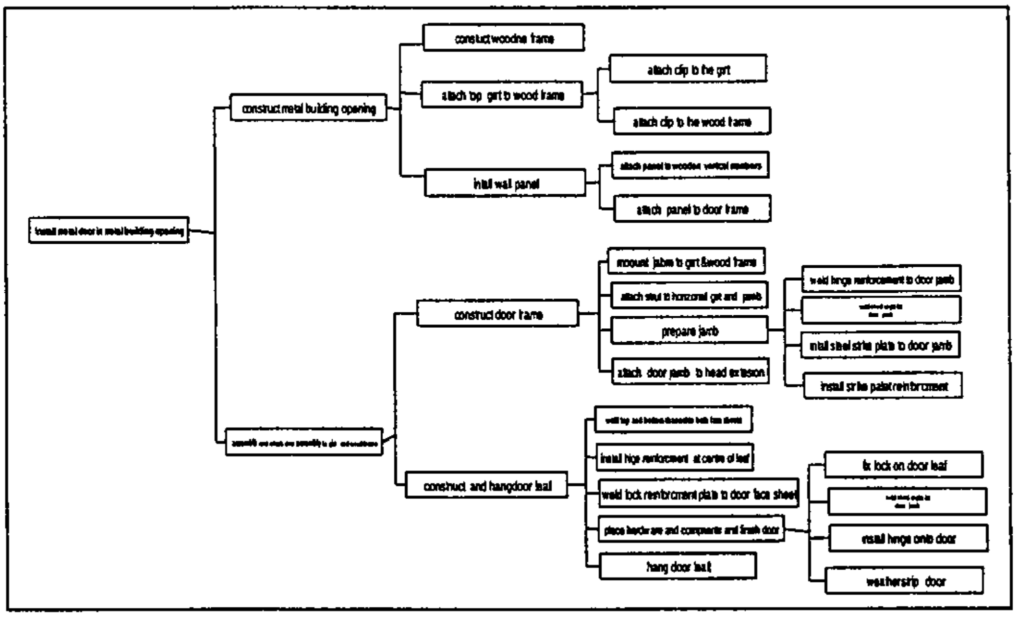

install metal door in metal building opening
construct metal building opening
construct wooden frame
attach top girt to wood frame
attach clip to the girt
attach clip to the wood frame
install wall panel
attach panel to wooden vertical members
attach panel to door frame
construct door frame
mount jabm to girt &wood frame
attach stud to horizontal girt and jamb
prepare jamb
attach door jamb to head extension
weld hinge reinforcement to door jamb
install steel strike plate to door jamb
install strike palat reinforcement
construct and hangdoor leaf
weld top and bottom ... both face sheets
install hige reinforcement at centre of leaf
weld lock reinforcement plate to door face sheet
place hardware and components and finish door
hang door leaf
fix lock on door leaf
install hinge onto door
weatherstrip door

Figure 8.3 Detailed view of the "metal door in a metal opening of a building project" construction process

8.3.1 Install Metal Door in a Metal Building Opening

The "install metal doors in a metal building opening" process is a high level process, which can be accomplished through more detailed processes. This process describes the way in which this type of doors may be installed in buildings with metal openings The high level process " Construct Metal Door in a Metal Building Opening" is decomposed into two sub-processes. These sub-processes in the order of occurrence are: construct metal building opening and assemble and attach door assembly to horizontal girt and wood frame of the opening These sub-processes may be accomplished in various ways as illustrated in the further sections.

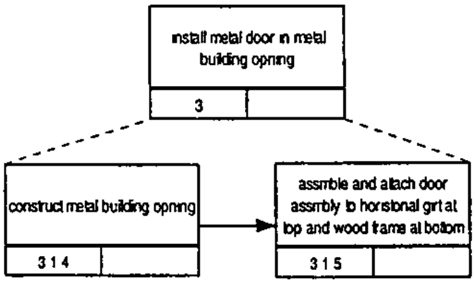

install metal door in metal building opning
3
construct metal building opning
314
assmble and attach door assmbly to horistonal girt at top and wood frame at bottom
315

Figure 8.4 Install Metal Door in a Metal Building Opening Process

Figure 8.4 shows the decomposition of the "Install Metal Door in Metal Building Opening" process Each of these sub-processes are related to another via, a precedes arrow This indicates that the process on the left must be completed before the commencement of the task on the right. For further Information on the process representation concepts and notations of IDEF3 Process Description Capture Method used in the representation of these processes, there is literature and references on this topic in **chapter 2** of this thesis

8.3.2 Construct Metal Building Opening

The installation of the door assembly in the building opening requires the construction of the opening on which it is attached. The "Construct Metal building Opening" shown in **figure 8.5,** is similar to its parent process in that the decomposition of this process results in three ordered more detailed sub processes linked via precedes relation arrows. These sub-processes define the building opening construction process and these sub-processes in their order of occurrence are: construct wooden frame, install girt or attach top girt to wooden frame, and install wall panel

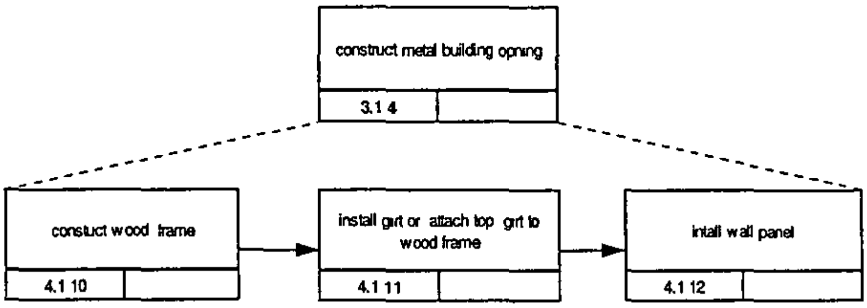

construct metal building opning
3.1 4
constuct wood frame
4.1 10
install girt or attach top girt to wood frame
4.1 11
intall wall panel
4.1 12

Figure 8.5 Construct Metal building Opening Process

The two processes represented by boxes 4.1 11 and 4.1.12 are processes that may need to be accomplished via other lower level processes Theses two sub-processes are decomposed and detailed in further sections The "Construct Wood Frame" is an activity at the lowest level of decomposition of the "Construct Metal building Opening" process. The process information related to design, the materials used and products produced, which are extracted from the case study project construction specification (shown in table **8.1** & **8.2**) for the "Construct Wooden Frame task" activity are as listed below.

□ Construct Wooden Frame task

- requires: 2*12 Douglas Fire Wood
- produces· 97"w*102 ½"h Wooden Frame

8.3.3 Install Girt or Attach Top Girt to Wood Frame

The "Install Girt or Attach Top Girt to Wood Frame" Process is the second step in the parent "Construct Metal building Opening" process. This process is decomposed into two distinct activities, which are. "attach clip to girt" and "attach clip to wooden frame" by which the installation of the girt is accomplished After these activities are completed the wall panel can be installed.

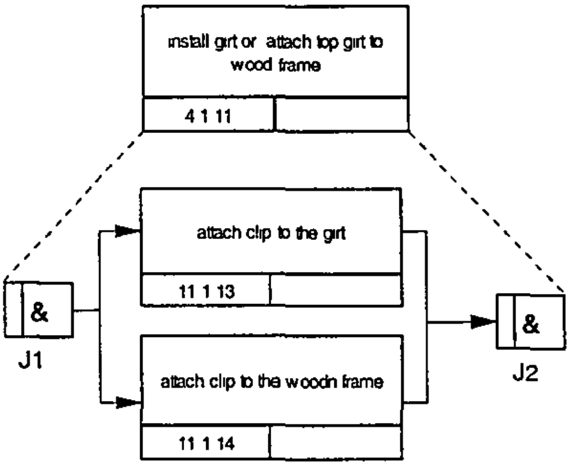

install girt or attach top girt to wood frame
4 1 11
attach clip to the girt
11 1 13
attach clip to the woodn frame
11 1 14
&
J1
&
J2

Figure 8 6 Install Girt or Attach Top Girt to Wood Frame Process

Figure 8.6 depicts a scenario in which the activities in boxes 11.1 13 and 11 1.14 can be initiated and completed in any order of time as illustrated by the IDEF3 asynchronous junctions J1 and J2 The following lists the process information related to design; the materials used and products produced, as extracted from the case study project construction specification (shown in **tables 8.1 & 8.2**) for the “attach clip to girt” and “attach clip to wooden fame” activities.

□ Attach clip to girt task

- Requires: 8” girt with 3” flange, six #12*1 ¼” Self Drilling Screws, 0 563”steel washers
- produces: clip on girt

□ Attach clip to wooden frame task

- requires four #12 *1 ½”Wood Lag Screws
- produces clip attached to girt; girt attached to wooden frame with clips.

8.3.4 Install Wall Panel

The “Install Wall Panel” Process is the third step in the parent “Construct Metal building Opening” Process. This process decomposes into two sub-processes by which the parent process may be completed These sub-processes “Install Wall Panel or Cover Portion of Opening”, and “Attach Panel to Wooden Vertical Members” are activities at the lowest level of decomposition of the patent process “Install Wall Panel”. **Figure 8.7** shown the process in which these two activities may start and complete in any order of time as indicated by the IDEF3 asynchronous junctions J1 and J2. The performance of these activities within the decomposition of the parent process depends on the fact that the vertical members and the wood frame are constructed and completed

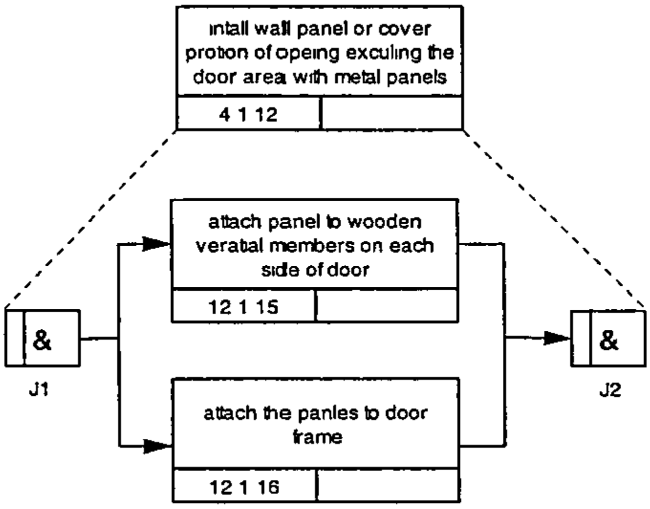

intall wall panel or cover proton of opeing exculing the door area with metal panels
4 1 12
attach panel to wooden veratal members on each side of door
12 1 15
attach the panles to door frame
12 1 16
&
J1
&
J2

Figure 8.7. Install Wall Panel Process

Process information related to design; the materials used and products produced, as extracted from the case study project construction specification (shown in **tables 8.1 & 8.2**) for the "attach clip to girt" and "attach clip to wooden fame" activities are as follows.

□ Attach panel to wooden vertical members on each side of door task

- requires. wall panel, twenty two #12*1 ¼" Self drilling Screws,
- produces: wall panel attached to vertical members

□ Attach Panel to door frame

- requires nine #12*1 ¼" Self drilling Screws
- produces: wall panel attached to door frame

8.3.5 Attach Door Assembly to Girt and Wood Frame of the Building Opening

The "Attach Door Assembly to Girt and Wood frame of the building opening" process defines the way in which the door should be constructed and attached to the building opening. This process is accomplished through the construction of the doorframe and leaf before the hardware and components can be fixed to the door. **Figure 8.8** shows the decomposition of the process into two sub processes. These processes need to be accomplished through other lower lever sub-processes. The "Attach Door Assembly to Girt and Wood frame" process is the second step in the high-level "Install Metal Door in

Metal Buıldıng Openıng" process. Once the construction of the door openıng ıs completed these two sub processes can commence asynchronously. The IDEF3 Junctıons J1 and J2 ın **figure 8.8** convey an asynchronous relatıonshıp between the constructıon of doorframe and door leaf processes that they can start and fınısh in any order of tıme.

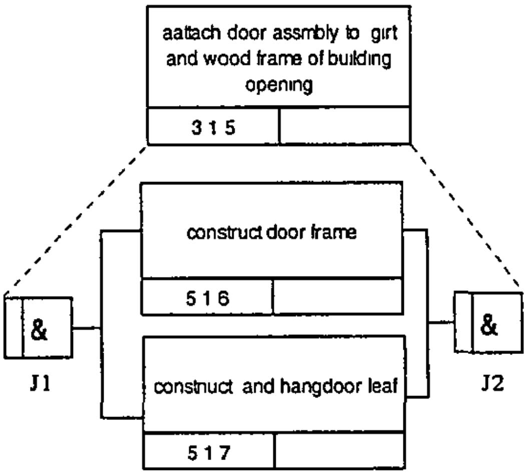

aattach door assmbly to girt and wood frame of buıldıng openıng
3 1 5
construct door frame
5 1 6
&
J1
&
J2
consnuct and hangdoor leaf
5 1 7

Figure 8.8. Attach Door Assembly to Gırt and Wood frame of the buıldıng openıng Process

8.3.6 Construct Door Frame

The "Construct Door Frame" process creates a frame to whıch the door can be hınged. The accomplıshment of this process requıres the completıon of each of the four sub-processes shown ın **figure 8.9**. Whıle the sub-processes represented by boxes 6.1.17, 6 1.18 and 6 1.20 are actıvıties at the lowest level of detaıl of the parent process, the sub-process ın box 6.1.19 requıres further decomposıtıon ınto more detaıled sub-process by whıch thıs process can be accomplıshed The arrows between the boxes sıgnıfy sımple precedence relatıonshıp between the four sup-processes.

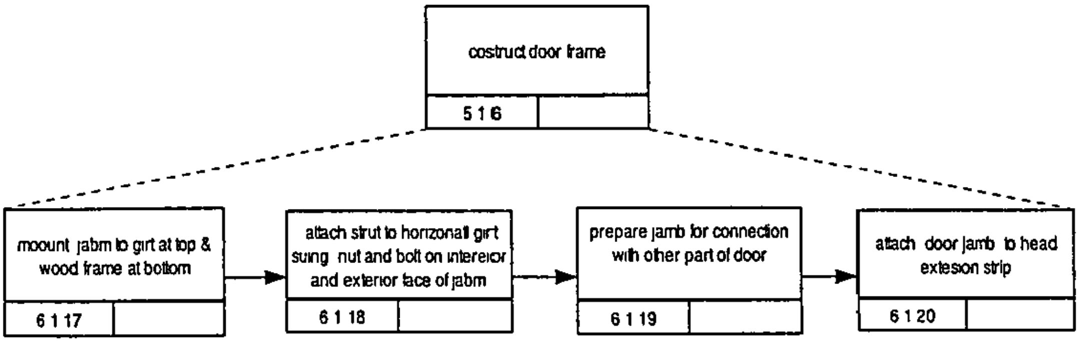

costruct door frame
5 1 6
moount jabm to girt at top & wood frame at bottom
6 1 17
attach strut to horizonall girt suing nut and bolt on intereior and exterior face of jabm
6 1 18
prepare jamb for connection with other part of door
6 1 19
attach door jamb to head exteson strip
6 1 20

Figure 8.9. Construct Door Frame Process

The "Mount doorjamb to girt and wood frame task" "Attach strut to horizontal girt and doorjamb task" and "Attach door jamb to head extension strip" sub-processes are activities that can be found at the lowest level of decomposition of the parent process "Construct Door Frame", and the process information for each of these activities related to design; the materials used and products produced, as extracted from the case study project construction specification (shown in **tables 8.1 & 8.2)** are as follows.

□ Mount doorjamb to girt and wood frame task

- requires: doorjamb, four ¼"*1 machine screws, four 5/6"*1 5/8" wood lags
- produces door jamb in place

□ Attach strut to horizontal girt and doorjamb task

- requires· sliding 1 ¼"*8"h galvanized steel strut , two 5/6" X ¾" nut and bolt
- produces: girt and doorjamb fastened in place with bolts and nuts

□ Attach door jamb to head extension strip

- requires· two 5/16" x ¾" hex head nut and bolt screws
- produces doorjamb attach to head

8.3.7 Prepare Doorjambs

The "Prepare Doorjambs" is a high level process, which can be accomplished via other lower level processes shown in **figure 8.10** This representation illustrates that preparation of a doorjambs for this type of door require four distinct sub processes, which are: "weld hinge reinforcement" and "weld steel angle" and "install strike plate" and "install strike plate reinforcement" to doorjambs. This process puts all the necessary reinforcements for hinges and strike plates before the hinges and plates can be fixed to doorjambs As described for other processes, these four sub-processes can start and complete in any order of time as indicated by the IDEF3 Process Description Method asynchronous junctions J1 and J2 of **figure 8.10** Once these four sub-processes are completed the doorjambs can be attached to head extension before the door leaf can be fixed to the jambs.

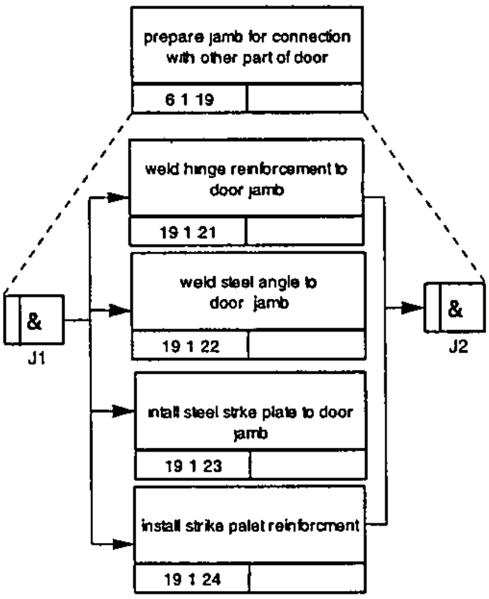

prepare jamb for connection with other part of door
6 1 19
weld hinge reinforcement to door jamb
19 1 21
weld steel angle to door jamb
19 1 22
intall steel strke plate to door jamb
19 1 23
install strike palet reinforcment
19 1 24
&
J1
&
J2

Figure 8.10. Prepare Doorjambs Process

The process information related to design; the materials used and products produced, as extracted from the case study project construction specification (shown in **tables 8.1 & 8.2**) for the "weld hinge reinforcement" and "weld steel angle" and "install strike plate" and "install strike plate reinforcement" to doorjambs sub-processes, which are activities that can be found at the lowest level of decomposition of the parent process "Prepare Doorjambs", are as follows

□ Weld hinge reinforcement to doorjamb task

- requires. doorjamb;-three, 7 Gage, 10" x 1 ¼"" hinge reinforcement
- produces: hinge reinforcements welded to doorjamb

□ Weld steel angle to doorjamb

- requires· 12 Gage galvanized steel angle measuring 1 ¾" x2" x7 ½" x 0 099"thick. 1 ¾" leg
- produces: steel angle welded to doorjamb

□ Install steel strike plate to doorjamb task

- requires: 4.86" x 1.75" x0 079" thick steel strike plate
- produces· strike plate welded to doorjamb
  - □ Install strike plate reinforcement
- requires three point spot welded 16 gage x 1 ¼" x6" strike plate reinforcement
- produces reinforcements for strike plate

8.3.8 Construct and Hang Door Leaf

The "Construct and Hang Door Leaf" process is accomplished via three sub-processes represented by boxes 7 1.25, 7.1.26 and 7.1 27, which can commence and complete in any order of time, as indicated by the IDEF3 Process Description Method asynchronous junctions J1 & J2, before the hardware and components are placed and the door leaf is hanged via the "place hardware and components" and "hang door leaf" processes as illustrated in **figure 8.11** While the sub-processes in boxes 7 1.25, 7.1.26, 7.1.27, and 7.1 29 are activities that don't need further decomposition for completion for each of these, the sub-process in box 7.1 28 requires further decomposition into more detailed activities, by which the completion of the "hardware and component" process can be met. **Figure 8.11** shows that process box 7.1 28 cannot commence until all the three asynchronous activities are completed. The performance of the three activities within the decomposition of the parent process doesn't depend on any other process. It can be initiated or completed at any time that is before, parallel or after the "construct door opening" or " construct doorframe" processes.

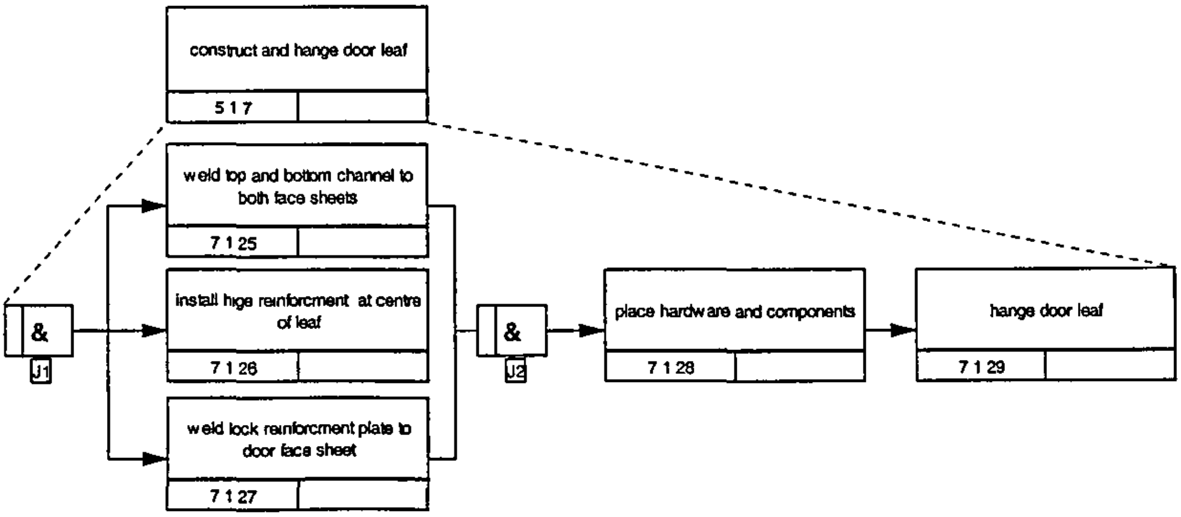

construct and hange door leaf
5 1 7
weld top and bottom channel to both face sheets
7 1 25
&
J1
install hige reinforcment at centre of leaf
7 1 26
weld lock reinforcment plate to door face sheet
7 1 27
&
J2
place hardware and components
7 1 28
hange door leaf
7 1 29

Figure 8.11. Construct and Hang Door Leaf Process

The "Weld top and bottom channels to both face sheets of door leaf", "Install hinge reinforcement at centre of door leaf", "Weld lock reinforcement plate to door leaf" and "hang door leaf" activities have the following process information related to design; the materials used and products produced, as extracted from the case study project construction specification (shown in **tables 8.1 & 8.2**).

□ Weld top and bottom channel to both faces sheets of door leaf

- requires door leaf with Expanded polystyrene core, 16 Gage flush top and bottom channel
- produces door leaf with channel welded to both face sheets

□ Install hinge reinforcement at centre of the leaf

- requires. door leaf with Expanded polystyrene core, (3) 9" x 1 ¼" x 7 Gage hinge reinforcement
- produces: door leaf with hinge reinforcement

□ Weld a steel lock reinforcement plate to door face sheet

- requires door leaf with Expanded polystyrene core; 16 Gage steel lock reinforcement plate
- produces lock reinforcement plate welded to door leaf.

□ Hang door on door frame

- requires· screws
- produces. door leaf hanged on door frame

8.3.9 Place Hardware and Components and Finish Door

The "Place Hardware and Components onto Door" process is the final step in the overall process of the cast study scenario project. This process should be accomplished via three activities. These are· "fix lock on door leaf ", "install hinge on door leaf" and "apply weatherstrip to door" asynchronous activities. **Figure 8.12** illustrates the asynchronous behavior of this process in which the three activities can start and complete in any order of time, which is conveyed by the IDEF3 Process Description Method asynchronous J1 and J2 junctions

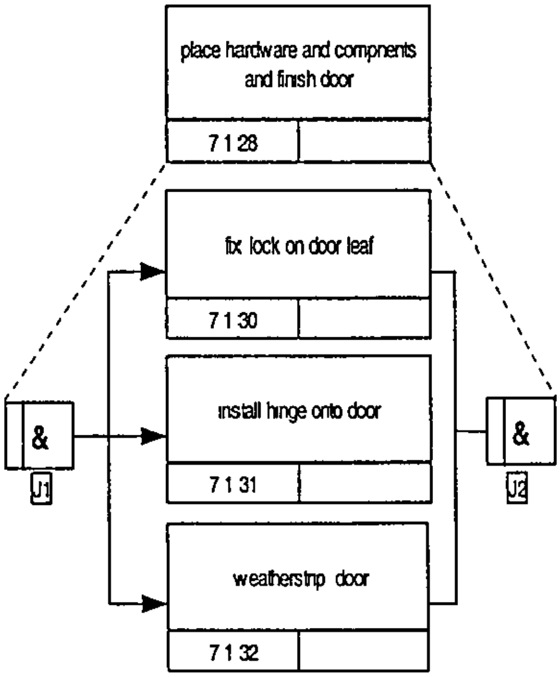


Figure 8.12. Place Hardware and Components and Finish Door Process

The following lists show the process information related to design for each of the three activities; the materials used and products produced, as extracted from the case study project construction specification (shown in **tables 8.1 & 8.2**).

□ Fix lock on door leaf

- requires: (1) Yale 5307 Lever Lock
- produces: lock fixed on door leaf

□ Install hinge on door leaf

- requires: 4 ¼" steel Hinges with non-removable pin template
- produces: hinge installed on door leaf

□ weatherstrip door

- requires AMSCO extruded aluminum & vinyl weatherstrip
- produces: weatherstripped door

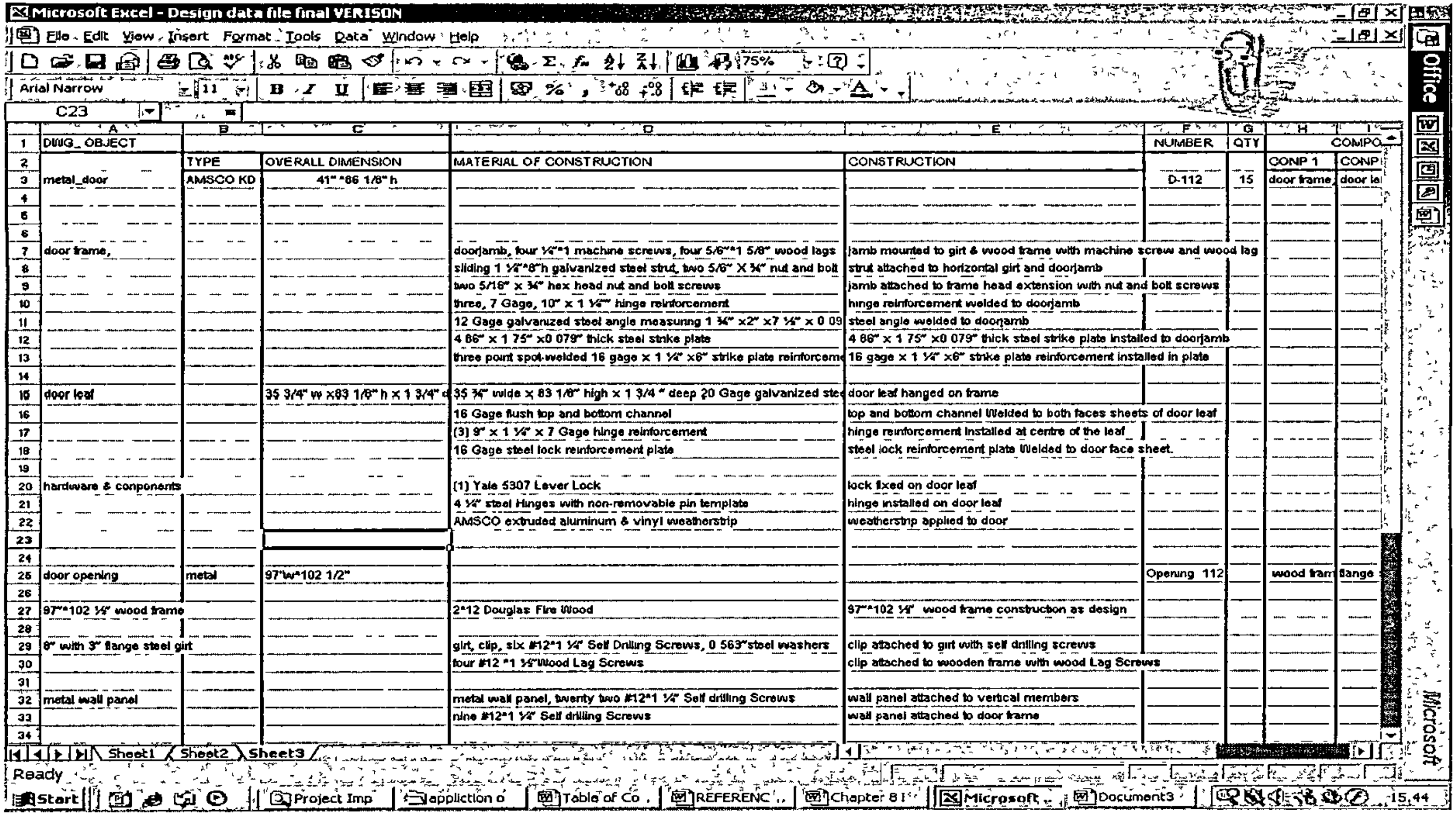
8.4 Process Information Representations using Scenario Applications

In the previous section the on site construction processes for the case study project were modeled using IDEF3 Process Description Capture Method and data related to design stage of the scenario, which were extracted from the case study project construction specification (shown in tables 8.1 & 8 2) were attached to the model at the activity level of decomposition of the processes. This referred to as process information related to design. In order to develop the full construction process information related to the scenario that is process information related to design, pre-tender cost estimating, and scheduling for the case study project, it was determined to use the scenario software applications' information representation files. These applications provide data fields for the representation of the processes and activities of the model and associated data related to each process and activity In this way, the applications representations of the process information provided a view into the data types of the applications representing the case study project information, hence a view into the native ontology of each of the applications. The native ontology of each of the applications is important in the translation process, because this ontology is mapped into the PSL ontology and information translation is conducted between these applications' native ontologies and PSL ontology, therefore the translation definitions must be written between these ontologies and the PSL ontology Additionally the use of the scenario software applications provided a clear view into the applications' data types and the concepts represented by these data types Hence an understanding of the use of the same or different data types to represent different or the same process concepts is achieved during the applications' representations of the case study project construction process information. This provided the shared information between those applications that need to be exchanged between them during the pre-construction process stages.

As described above although these applications have different file formats there are some common exchange file formats that can be used to export and import files between the three scenario software applications. CCS and Microsoft project have a default data format that is the same for all types of projects and for all users, however for AutoCAD there is no such data format and every designer or user needs to structure the information that can be extracted from design drawings In the following sections, a brief description of the scenario software applications representations of the case study project construction process information is presented.

8.4.1 AutoCAD Design Data Representation using Excel File Format

AutoCAD drawing objects provide geometrical representation of the case study project products to be realized by the construction processes of the IDEF3 model. A drawing representation of the case study project is attached in **appendix C** Attributes that can be attached to AutoCAD drawing objects are given in the case study project construction specification as shown in **tables 8.1 & 8.2.** In AutoCAD these specifications can be attached to drawing objects and extracted using AutoCAD tools for use in a spreadsheet or database files to produce textual information structured in file formats that can be exported and imported to the other two scenario applications. **Figure 8.13** shows a screen shot of the textual information, extracted from the case study project construction specification shown in **tables 8.1 & 8.2**, in relation to the drawing objects of the case study project, structured in Excel file format These are construction case study project objects represented using drawing format, including, their component parts and other process related information. The drawing objects at the highest level have data fields for ID, quantity, and component parts etc of the products. The component parts of the drawing objects at the lowest level are drawing objects on their own rights and have data fields associated with them for methods and material of construction etc In this section excel file format is used to structure the textual information associated with the drawing objects of the cases study project. This file format allows for the information to be exported and imported to and from CCS and Microsoft project scheduling applications. The design related textual construction process information of the case study project is organized and structured using excel file as shown in **figure 8.13**

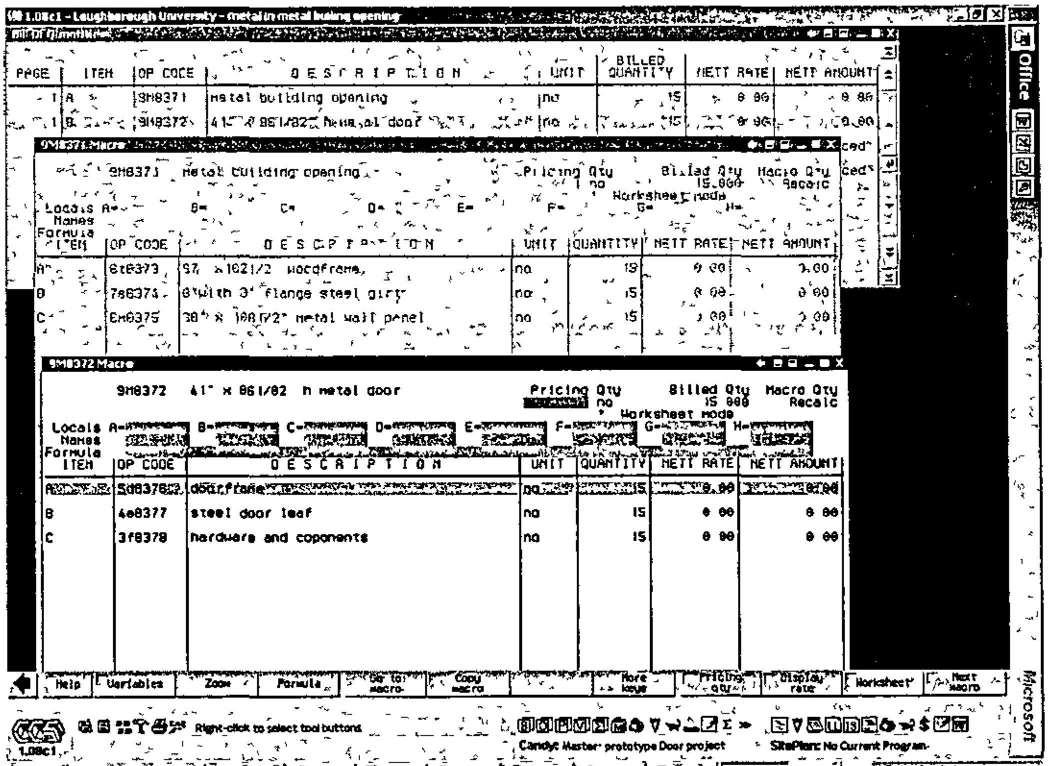


	A	B	C	D	E	F	G	H	I
1	DWG_ OBJECT					NUMBER	QTY		COMPO
2		TYPE	OVERALL DIMENSION	MATERIAL OF CONSTRUCTION	CONSTRUCTION			CONP 1	CONP
3	metal_door	AMSCO KD	41" *86 1/8" h			D-112	15	door frame	door le
4									
5									
6									
7	door frame,			doorjamb, four ¼"*1 machine screws, four 5/6"*1 5/8" wood lags	jamb mounted to girt & wood frame with machine screw and wood lag				
8				sliding 1 ¼"*8"h galvanized steel strut, two 5/6" X ¾" nut and bolt	strut attached to horizontal girt and doorjamb				
9				two 5/16" x ¾" hex head nut and bolt screws	jamb attached to frame head extension with nut and bolt screws				
10				three, 7 Gage, 10" x 1 ¼"" hinge reinforcement	hinge reinforcement welded to doorjamb				
11				12 Gage galvanized steel angle measuring 1 ¾" x2" x7 ½" x 0 09	steel angle welded to doorjamb				
12				4 86" x 1 75" x0 079" thick steel strike plate	4 86" x 1 75" x0 079" thick steel strike plate installed to doorjamb				
13				three point spot-welded 16 gage x 1 ¼" x6" strike plate reinforceme	16 gage x 1 ¼" x6" strike plate reinforcement installed in plate				
14									
15	door leaf		35 3/4" w x83 1/8" h x 1 3/4" d	35 ¾" wide x 83 1/8" high x 1 3/4 " deep 20 Gage galvanized stee	door leaf hanged on frame				
16				16 Gage flush top and bottom channel	top and bottom channel Welded to both faces sheets of door leaf				
17				(3) 9" x 1 ¼" x 7 Gage hinge reinforcement	hinge reinforcement installed at centre of the leaf				
18				16 Gage steel lock reinforcement plate	steel lock reinforcement plate Welded to door face sheet.				
19									
20	hardware & conponents			(1) Yale 5307 Lever Lock	lock fixed on door leaf				
21				4 ½" steel Hinges with non-removable pin template	hinge installed on door leaf				
22				AMSCO extruded aluminum & vinyl weatherstrip	weatherstrip applied to door				
23									
24									
25	door opening	metal	97"w*102 1/2"			Opening 112		wood fram	flange
26									
27	97"*102 ½" wood frame			2*12 Douglas Fire Wood	97"*102 ½" wood frame construction as design				
28									
29	8" with 3" flange steel girt			girt, clip, six #12*1 ¼" Self Drilling Screws, 0 563"steel washers	clip attached to girt with self drilling screws				
30				four #12 *1 ½"Wood Lag Screws	clip attached to wooden frame with wood Lag Screws				
31									
32	metal wall panel			metal wall panel, twenty two #12*1 ¼" Self drilling Screws	wall panel attached to vertical members				
33				nine #12*1 ¼" Self drilling Screws	wall panel attached to door frame				
34									

Figure 8.13. design data of the case study project structured in Excel file format

8.4.2 Cost Estimating Data Representation using CCS Files

The CCS cost estimating application uses ASCII text files to represent cost estimating related process information. CCS uses bill of quantities (BoQ), macros, and operations files, at the different level of abstractions to organize the processes and products (input data from design stage) and provide data fields for other cost estimating related data. CCS uses separate worksheets files to calculate nett-rates for each operation based on suppliers' resources cost rates Operations in CCS are attached to worksheet where the resources such as material, labour, plant are determined and the suppliers' rates of the resources that may be consumed or/and used by each operation are identified and used to produce nett-rates per billed quantity of each operation. The nett-rate and billed quantity of each operation produces the nett-amount of the operation Using the operations' nett-rates the value of other items (the macros or operation) at the higher level of abstraction such as the nett-rates and nett-amounts, can be calculated and finally combined to form the overall project cost Figures, 8 14 (a) & (b) show screen shots of CCS application's bill of quantity BoQ, and macro files at different level of abstraction; and the operations and worksheet files respectively that are used to organize the cost estimating related process information of the case study project.



1.08c1 - Loughborough University - metal in metal building opening

PAGE	ITEM	OP CODE	DESCRIPTION	UNIT	BILLED QUANTITY	NETT RATE	NETT AMOUNT
1	A	9H8371	metal building opening	no	15	0 00	0 00
1	B	9H8372	41" x 861/82 h metal door	no	15	0 00	0.00

9M8371 Macro

9H8371 Metal building opening — Pricing Qty 1 no — Billed Qty 15.000 — Macro Qty Recalc — Worksheet mode

Locals Names Formula A= B= C= D= E= F= G= H=

ITEM	OP CODE	DESCRIPTION	UNIT	QUANTITY	NETT RATE	NETT AMOUNT
A	6t8373	37 x 102 1/2 woodframe,	no	15	0 00	0.00
B	7s8374	8"with 3" flange steel girt	no	15	0 00	0 00
C	6m8375	30" x 100 1/2" metal wall panel	no	15	0 00	0 00

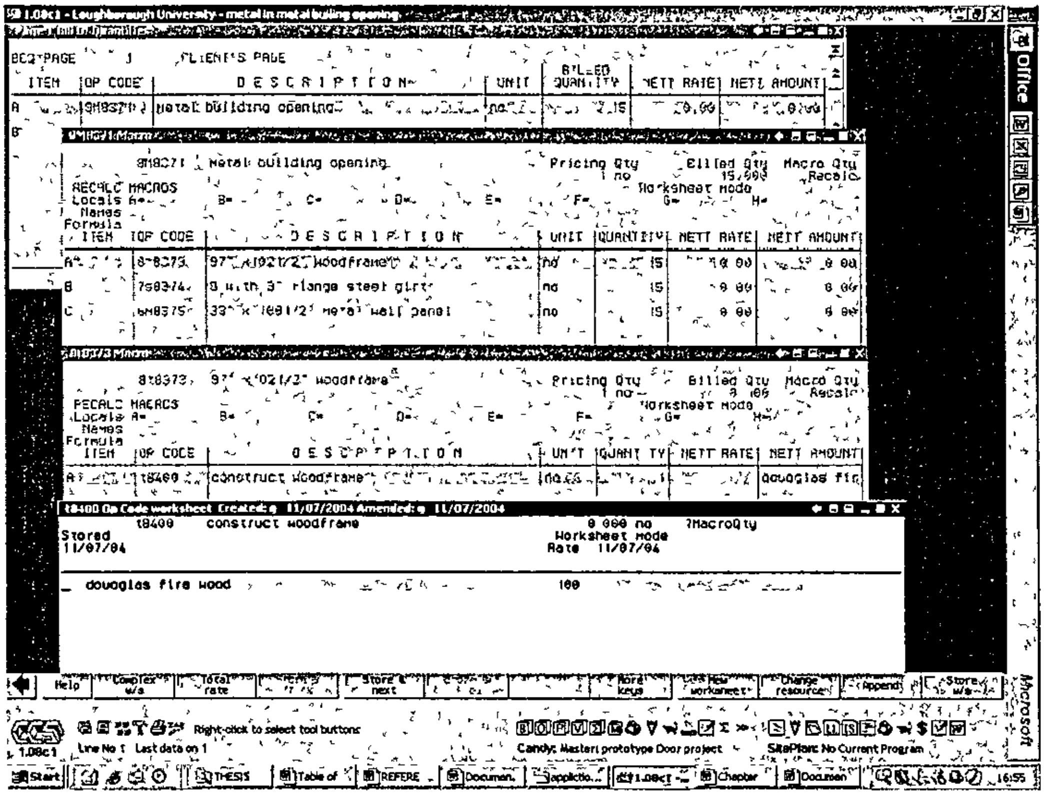
9M8372 Macro

9H8372 41" x 861/82 h metal door — Pricing Qty no — Billed Qty 15 000 — Macro Qty Recalc — Worksheet mode

Locals Names Formula A= B= C= D= E= F= G= H=

ITEM	OP CODE	DESCRIPTION	UNIT	QUANTITY	NETT RATE	NETT AMOUNT
A	5d8376	doorframe	no	15	0.00	0.00
B	4e8377	steel door leaf	no	15	0 00	0 00
C	3f8378	hardware and coponents	no	15	0 00	0 00

Figure 8.14. (a) cost estimating data of the case study project structured in CCS BoQs and Macro files



1.08c1 - Loughborough University - metal in metal builing opening

BCQ PAGE 1 CLIENT'S PAGE

ITEM	OP CODE	DESCRIPTION	UNIT	BILLED QUANTITY	NETT RATE	NETT AMOUNT
A	9M8371	metal building opening	no	15	20.00	[illegible]
B						

9M8371 Metal building opening Pricing Qty 1 no Billed Qty 15.000 Macro Qty Recalc
RECALC MACROS Worksheet mode
Locals A= B= C= D= E= F= G= H=
Names
Formula

ITEM	OP CODE	DESCRIPTION	UNIT	QUANTITY	NETT RATE	NETT AMOUNT
A	8-8373	9'-4 1/2" Woodframe	no	15	10.00	0.00
B	7s8374	8 with 3" flange steel girt	no	15	0.00	0.00
C	6M8375	33" x 108 1/2" metal wall panel	no	15	0.00	0.00

8t8373 9'-4 1/2" woodframe Pricing Qty 1 no Billed Qty 0.100 Macro Qty Recalc
RECALC MACROS Worksheet mode
Locals A= B= C= D= E= F= G= H=
Names
Formula

ITEM	OP CODE	DESCRIPTION	UNIT	QUANTITY	NETT RATE	NETT AMOUNT
A	t8400	construct woodframe	no	[illegible]	[illegible]	douoglas fir

t8400 Op Code worksheet Created: 11/07/2004 Amended: 11/07/2004

t8400 construct woodframe 0.000 no ?MacroQty
Stored Worksheet mode
11/07/04 Rate 11/07/04

douoglas fire wood 100

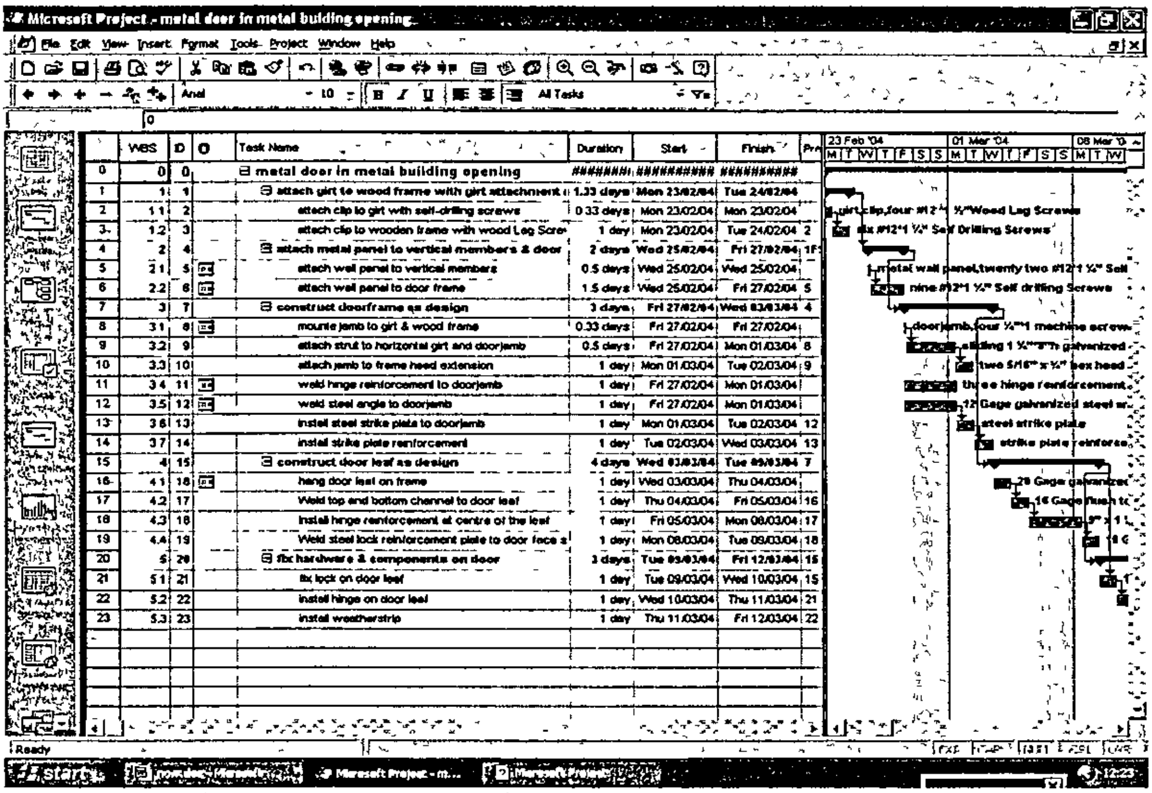
Line No 1 Last data on 1
Candy: Master prototype Door project SitePlan: No Current Program

Figure 8.14 (b) cost estimating data of the case study project structured in CCS Operation & work sheet files

8.4.3 Construction Plan Representation using Microsoft Project Files

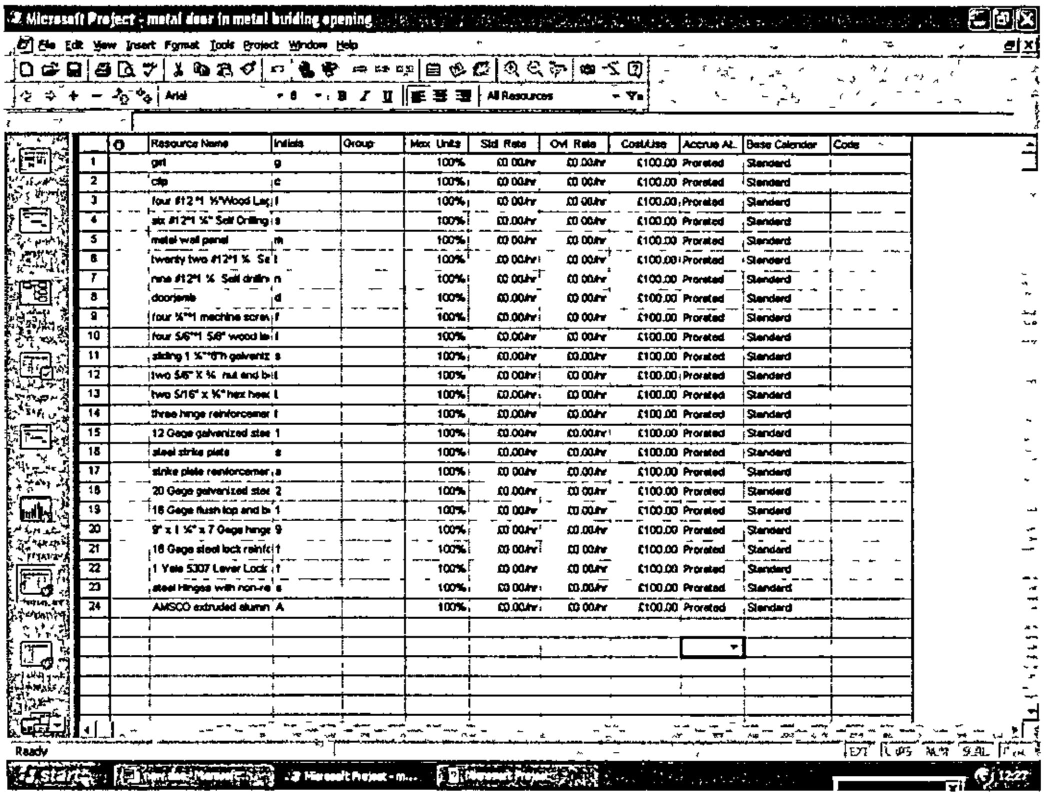
Microsoft project scheduling application uses Gantt charts and Resource Sheets to structure processes and activities, resources and other related information A typical project consists of a series of interrelated tasks, the building blocks of a schedule In Microsoft project the processes at different level of abstraction, and information related to the processes and activities are structured into summary tasks and sub-tasks and associated data fields for resources and other scheduling information in the Gantt chart Resource Sheets provide the capability for creating a resource list and associated information for a project, such as resource name, groups, costs etc. any of these resources can be assigned to tasks, with associated data fields for different types of cost rates and cost values, when these resources are assigned to tasks Microsoft project automatically calculates the cost of the tasks Microsoft project application's capability of process information representation is not limited to these scheduling concepts described in this section however the focus of this project is on the process information that are shared between AutoCAD and CCS applications and limited to the concepts described. **Figures 8.15 (a & b)** show screen shots of the Microsoft project's Gantt charts and Resource

Sheets representatıons of the processes and resources and assocıated ınformatıon for the case study project constructıon.


Microsoft Project - metal door in metal building opening
File Edit View Insert Format Tools Project Window Help
Arial 10 All Tasks
23 Feb '04
01 Mar '04
08 Mar '04
M T W T F S S M T W T F S S M T W
Ready

	WBS	ID	O	Task Name	Duration	Start	Finish	Pre
0	0	0		⊟ metal door in metal building opening	########	##########	##########	
1	1	1		⊟ attach girt to wood frame with girt attachment	1.33 days	Mon 23/02/04	Tue 24/02/04	
2	1.1	2		attach clip to girt with self-drilling screws	0.33 days	Mon 23/02/04	Mon 23/02/04	
3	1.2	3		attach clip to wooden frame with wood Lag Scre	1 day	Mon 23/02/04	Tue 24/02/04	2
4	2	4		⊟ attach metal panel to vertical members & door	2 days	Wed 25/02/04	Fri 27/02/04	1F
5	2.1	5	▤	attach wall panel to vertical members	0.5 days	Wed 25/02/04	Wed 25/02/04	
6	2.2	6	▤	attach wall panel to door frame	1.5 days	Wed 25/02/04	Fri 27/02/04	5
7	3	7		⊟ construct doorframe as design	3 days	Fri 27/02/04	Wed 03/03/04	4
8	3.1	8	▤	mounte jamb to girt & wood frame	0.33 days	Fri 27/02/04	Fri 27/02/04	
9	3.2	9		attach strut to horizontal girt and doorjamb	0.5 days	Fri 27/02/04	Mon 01/03/04	8
10	3.3	10		attach jamb to frame head extension	1 day	Mon 01/03/04	Tue 02/03/04	9
11	3.4	11	▤	weld hinge reinforcement to doorjamb	1 day	Fri 27/02/04	Mon 01/03/04	
12	3.5	12	▤	weld steel angle to doorjamb	1 day	Fri 27/02/04	Mon 01/03/04	
13	3.6	13		install steel strike plate to doorjamb	1 day	Mon 01/03/04	Tue 02/03/04	12
14	3.7	14		install strike plate reinforcement	1 day	Tue 02/03/04	Wed 03/03/04	13
15	4	15		⊟ construct door leaf as design	4 days	Wed 03/03/04	Tue 09/03/04	7
16	4.1	16	▤	hang door leaf on frame	1 day	Wed 03/03/04	Thu 04/03/04	
17	4.2	17		Weld top and bottom channel to door leaf	1 day	Thu 04/03/04	Fri 05/03/04	16
18	4.3	18		Install hinge reinforcement at centre of the leaf	1 day	Fri 05/03/04	Mon 08/03/04	17
19	4.4	19		Weld steel lock reinforcement plate to door face s	1 day	Mon 08/03/04	Tue 09/03/04	18
20	5	20		⊟ fix hardware & components on door	3 days	Tue 09/03/04	Fri 12/03/04	15
21	5.1	21		fix lock on door leaf	1 day	Tue 09/03/04	Wed 10/03/04	15
22	5.2	22		install hinge on door leaf	1 day	Wed 10/03/04	Thu 11/03/04	21
23	5.3	23		install weatherstrip	1 day	Thu 11/03/04	Fri 12/03/04	22

Figure 8.15. (a) Gantt chart representatıon of the tasks and related ınformatıon for the case study project constructıon plan


Microsoft Project - metal door in metal building opening
File Edit View Insert Format Tools Project Window Help
Arial
8
All Resources
Ready
EXT CAPS NUM SCRL OVR
Start
Microsoft Project - m...

	ⓘ	Resource Name	Initials	Group	Max Units	Std Rate	Ovt Rate	Cost/Use	Accrue At	Base Calendar	Code
1		grit	g		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
2		clip	c		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
3		four #12*1 ½"Wood Lag	f		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
4		six #12*1 ¼" Self Drilling	s		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
5		metal wall panel	m		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
6		twenty two #12*1 ¼ Se	t		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
7		nine #12*1 ¼ Self drillin	n		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
8		doorjamb	d		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
9		four ¼"*1 machine screw	f		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
10		four 5/6"*1 5/8" wood la	f		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
11		sliding 1 ¾"*6'h galvaniz	s		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
12		two 5/8" X ¾ nut and bo	t		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
13		two 5/16" x ¾" hex head	t		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
14		three hinge reinforcemer	t		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
15		12 Gage galvanized stee	1		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
16		steel strike plate	s		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
17		strike plate reinforcemer	s		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
18		20 Gage galvanized stee	2		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
19		16 Gage flush top and b	1		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
20		9" x 1 ¾" x 7 Gage hinge	9		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
21		16 Gage steel lock reinfc	1		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
22		1 Yale 5307 Lever Lock	1		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
23		steel Hinges with non-re	s		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	
24		AMSCO extruded alumin	A		100%	£0.00/hr	£0.00/hr	£100.00	Prorated	Standard	

Figures 8.15 (b) Resource Sheets representation of the resources and related information for the case study project construction plan

8.5 Information Translation between Scenario Software Applications using PSL

In this section as an interchange language PSL is used to facilitate process information exchange between the scenario software applications; AutoCAD, CCS and Microsoft Project. The information translation between the applications files requires a two-stage process: syntactic and semantic translations. The syntactic translator is a parser between the PSL syntax (KIF notation) and the native syntax of each of the applications; this parser keeps the terminology of the application intact. The semantic translation substitutes terminology of each of the applications with the definitions written using PSL terminology The information exchange between the scenario applications is carried out through translation of information representations between each of the scenario software application's native ontology and PSL ontology using parsing and pattern matching (concepts mapping) procedures for syntactic and semantic translations respectively Hence in order to facilitate the mapping of information between the applications and PSL, the first step in the translation process is the development of each of the applications native ontology The ontology of each of the applications is hidden and is known by the developers only However, ontology of such applications can be developed from the user point of view that is, how an application structures information using the terminologies and file format, defined by the developers is the ontology of the application Based on this, the ontology of each of the scenario applications was developed from the case study project construction process information and the representation of this information using the applications terminologies (data types) and file formats as shown in **section 8.5.1, 8.5.5, 8.5.6 & 8.5.9.** In chapter 7, some conceptual information was gathered during the conceptual analysis and mapping of the scenario software applications data types (the applications' representations of process concepts) to PSL ontology. Based on this information, prior to the start of information translation process, hand written semantic translators (semantic translation definitions) were developed between the each of the applications and PSL ontology. **Figure 8.16** illustrates the case study project construction process information exchange between the three scenario applications and the role played by the PSL Ontology. **Figure 8.16** depicts the translation of process information representations between the three scenario software applications using Excel file as a common exchange file format for physical file exchange (import and export) between the applications; KIF notation for syntactic translation (parsing) of the applications' syntax to the PSL syntax (KIF notation); and

PSL as a common language for semantic translation between the applications' and PSL process information representation terminologies. This translation using PSL requires syntactic and semantic translators to translate information from the applications to PSL and back to the applications. While the syntactic translators are parsing procedures for syntactic translation between the applications information representation syntax and PLS syntax (KIF notation), the semantic translators are semantic translation definitions as shown in **section 8.5.1.** These translation definitions between an application ontology and PSL are driven by the ontological definitions that were written using the same foundational theories. These are definitions for the terminology of the application ontology, using *only* the terminology from the PSL Ontology, as well as definitions for the terminology of the PSL Ontology using *only* the terminology of the application ontology.

8.5.1 Semantic Translation Definitions for the Scenario Software Applications

Tra nslation Definition for AutoCAD Application (A)

```
(<=> (design-object@A ?r)
     (product-created  ?r))

(<=> (component@A ?r)
     (resource-created  ?r))

(<=> (material_of_construction @A ?r)
     (consumable  ?r))

(<=> (construction@A ?a)
     (processor_activity  ?a))
```

Translation Definition for CCS Application (C)

```
(<=> (bill_of_quantitiy_macro_op@C ?r)
     (product-created  ?r))

(<=> (macro_op@C  ?r)
     (resource-created  ?r))

   (<=> (operation@A ?a)
        (processor_activity  ?a))

   (<=> (material@C ?r)
        (consumable  ?r))

   (<=> (plant@C ?r)
        (possibly_reusable  ?r))

   (<=> (labour@C ?r)
        (reusable  ?r))
   (<=> (Nett_amount@C ?r)
        (activity_cost  ?r))

   (<=> (resource_amount@C ?r)
        (resource_cost  ?r))
```

Translation Definition for Microsoft Project Application (S)

```
(<=> (summary_task@C  ?r)
     (resource-created  ?r))

(<=> (task@S ?a)
     (processor_activity  ?a))

(<=> (resource@C ?r)
     (consumable  ?r))

(<=> (resource@C ?r)
     (possibly_reusable  ?r))

(<=> (resource@C ?r)
        (reusable  ?r))

(<=> (task_total_cost@C ?r)
     (activity_cost  ?r))
(<=> (resource_cost/use@C ?r)
     (resource_cost  ?r))
```

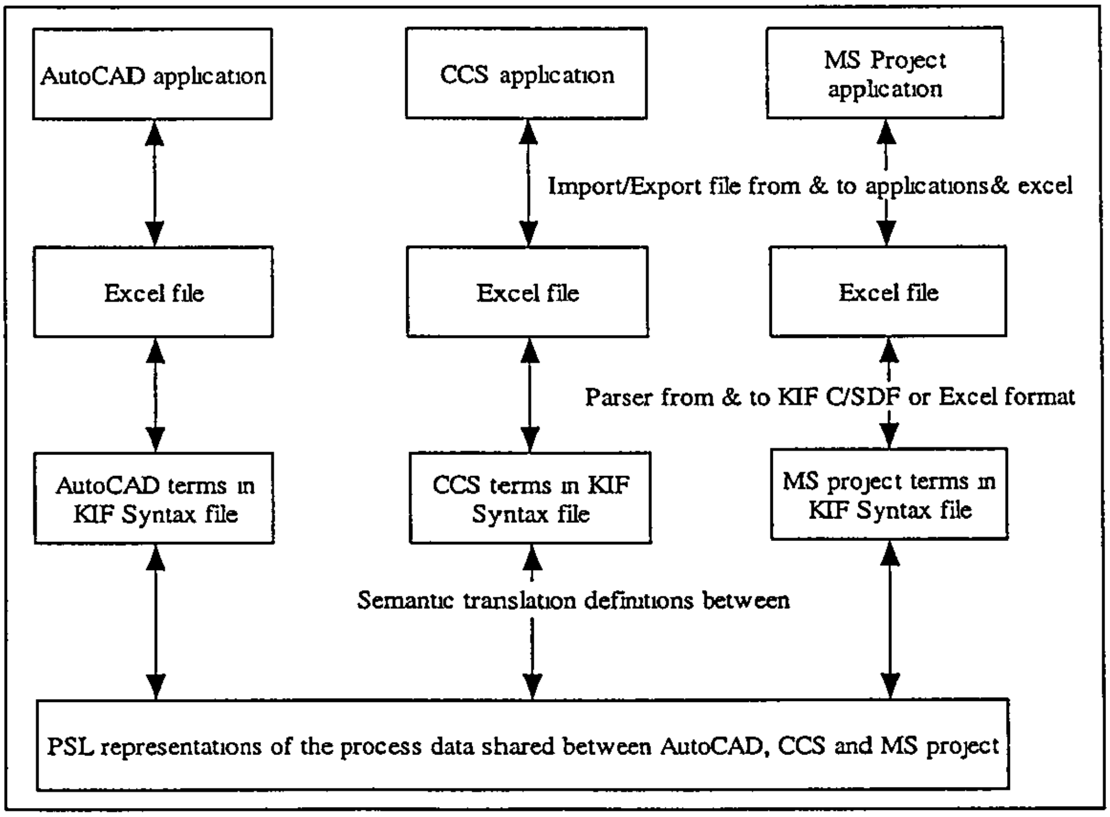

AutoCAD application
CCS application
MS Project application
Import/Export file from & to applications& excel
Excel file
Excel file
Excel file
Parser from & to KIF C/SDF or Excel format
AutoCAD terms in KIF Syntax file
CCS terms in KIF Syntax file
MS project terms in KIF Syntax file
Semantic translation definitions between
PSL representations of the process data shared between AutoCAD, CCS and MS project

Figure 8 16. Process information exchange between AutoCAD, CCS & Microsoft project applications using PSL as a common/standard language

8.5.2 Architectural Design Data File in AutoCAD Application's Representations

Architectural Design data (textual file) textual representation of AutoCAD application for a "metal door and a metal building opening" project Written in AutoCAD's textual attribute representation format and terminologies AutoCAD's textual attribute (information) representation formats are described **in chapter 6.**

{DWG-object	metal bldg opening (x)}
{components	wood frame; steel girt; wall panel }

{Component	97"*102 ½" wood frame metal bldg opening (x)}
{Material	2*12 Douglas Fire Wood (x)}
{construction.	wood-frame construction as design (x)}

{Component	8" with 3" flange steel girt metal bldg opening (x)}
{Material	girt, clip, six #12*1 ¼" Self Drilling Screws, 0 563"steel washers (x)}
{Construction	clip attached to girt with self-drilling screws (x)}
{Material.	four #12 *1 ½"Wood Lag Screws (x)}
{Construction	clip attached to wooden frame with wood Lag Screws (x)}

{ Component:	26 Gage 38"*108 ½"metal wall panel metal bldg opening (x)}
{Material	metal wall panel, twenty two #12*1 ¼" Self drilling Screws (x)}
{Construction·	wall panel attached to vertical members (x)}
{Material	nine #12*1 ¼" Self drilling Screws (x)}
{Construction	wall panel attached to door frame (x)}

{DWG-object	metal door (x)}
{components	doorframe, door leaf; hardware & components (x)}

{Component.	doorframe door assembly (x)}
{Material	doorjamb, four ¼"*1 machine screws, four 5/6"*1 5/8" wood lags (x)}
{Construction	jamb mounted to girt & wood frame with machine screw and wood lag (x)}
{Material	sliding 1 ¼"*8"h galvanized steel strut, two 5/6" X ¾" nut and bolt (x)}
{Construction.	strut attached to horizontal girt and doorjamb (x)}
{Material	two 5/16" x ¾" hex head nut and bolt screws (x)}
{ Construction	jamb attached to frame head extension with nut and bolt screws (x)}
{Material	three, 7 Gage, 10" x 1 ¼"" hinge reinforcement (x)}
{Construction	hinge reinforcement welded to doorjamb (x)}
{Material	12 Gage 1 ¾" x2" x7 ½" x 0 099"thick, 1 ¾" leg, galvanized steel angle (x)}
{ Construction	steel angle welded to doorjamb (x)}
{Material:	4.86" x 1.75" x0 079" thick steel strike plate (x)}
{Construction	4 86" x 1.75" x0 079" thick steel strike plate installed to doorjamb (x)}
{Material	three point spot-welded 16 gage x 1 ¼" x6" strike plate reinforcement (x)}
{Construction	16 gage x 1 ¼" x6" strike plate reinforcement installed in plate (x)}

{Component:	20 Gage galvanized steel door leaf door assembly (x)}
{Material·	35 ¾" wide x 83 1/8" high x 1 3/4 " deep 20 Gage galvanized steel door leaf (x)}
{Construction.	door leaf hanged on frame (x)}
{Material.	16 Gage flush top and bottom channel (x)}
{Construction	top and bottom channel Welded to both faces sheets of door leaf (x)}
{Material	(3) 9" x 1 ¼" x 7 Gage hinge reinforcement (x)}
{Construction	hinge reinforcement Installed at centre of the leaf (x)}
{Material·	16 Gage steel lock reinforcement plate (x)}
{Construction:	steel lock reinforcement plate Welded to door face sheet. (x)}

{Component·	hardware & components door assembly (x)}
{Material	(1) Yale 5307 Lever Lock (x)}

{Construction.	lock fixed on door leaf (x)}
{Material	4 ¼" steel Hinges with non-removable pin template (x)}
{ Construction	hinge installed on door leaf (x)}
{Material·	AMSCO extruded aluminum & vinyl weatherstrip (x)}
{Construction.	weatherstrip applied to door (x)}

8.5.3 Architectural Design Data File in PSL syntax (KIF Notation)

The syntactic translator takes/reads the architectural design data file in AutoCAD application's syntax and terminology (section 8.5.2) and produces a corresponding file using PSL syntax (KIF notation), but still persevering the AutoCAD application's terminology

```
(<=>(=>(metal_bldg-opening ?rx)
       (DWG_object ?x))
     (exists (?r1  ?r2 ?r3)
       (and (=> (wood_frame ?r1)
                (component   ?r1 ?x))
            (=> (8" with 3" steel_girt  ?r2)
                (component ?r2 ?x))
             (=> (wall_panel)
                 (component ?r3 ?x)))))


(<=> (=>(and (metal_bldg-opening ?x)
             (97"*102 ½" wood frame  ?r))
     (and (DWG_object ?x)
          (component  ?r ?x)))
     (exists (?a)
          (=>(wood-frame_constructed_as_design  ?a)
              (construction   ?a  ?r))
     (exists (?r1)
          (=>(2*12 Douglas Fire Wood  ?r1)
            (material_of_construction   ?r1 ?a)))))


(<=> (=>(and (metal_bldg-opening ?x)
             (8" with 3" flange steel girt  ?r))
         (and (DWG_object ?x)
                 (component  ?r ?x)))
     (exists (?a1 ?a2 )
              (and (=>(clip attached to girt ?a1)
                        (construction   ?a1  ?r))
                   (=>(clip attached to wooden frame ?a2)
                        (construction   ?a2  ?r)))
     (exists (?r1 ?r2 )
               (and (=>(girt, clip, six #12*1 ¼" Self Drilling
Screws ?r1)
                        (material_of_construction   ?r1 ?a1))
                    (=>(four #12 *1 ½"Wood Lag Screws  ?r2)
                        (material_of_construction   ?r2 ?a2))))))
```

```
(<=> (=>(and (metal_bldg-opening ?x)
             (26 Gage 38"*108 ½"metal panel  ?r))
         (and (DWG_object ?x)
             (component  ?r ?x)))
      (exists (?a1 ?a2 )
             (=>(and (=>(wall panel attached to vertical members
?a1)
                       (construction ?a1 ?r))
                    (=>(wall panel attached to door frame ?a2)
                       (construction ?a2 ?r)))
           (exists (?r1 ?r2 )
                (and (=>(metal panel,(22) #12*1 ¼" Self drilling
Screws ?r1)
                       (material_of_construction  ?r1  ?a1 ))
                    (=>(nine #12*1 ¼" Self drilling Screws?r2)
                       (material_of_construction   ?r2  ?a2))))))


(<=>(=>(metal door  ?x)
       (DWG_object ?r))
     (exists (?r1  ?r2 ?r3)
              (and (=> (door_frame ?r1)
                      (component   ?r1 ?x))
                      (=> (door_leaf  ?r2)
                      (component ?r2 ?x))
                    (=> (hard_ware_and_compenets ?r3)
                       (component ?r3 ?x)))))


(<=> (=>(and (metal door   ?x)
             (doorframe  ?r))
         (and (DWG_object ?x)
             (component  ?r ?x)))
    (exists (?a1 ?a2 ?a3 ?a4 ?a5 ?a6 ?a7)
            (and (=>(jamb mounted to girt & wood frame ?a1)
                       (construction   ?a1 ?r))
                (=>(strut attached to horizontal girt and
doorjamb ?a2)
                       (construction   ?a2 ?r))
                  (=>(jamb attached to frame head  ?a3)
                       (construction   ?a3 ?r))
                  (=>(hinge_reinforcement_welded_to_door_jamb
?a4)
                       (construction   ?a4 ?r))
                (=>(steel_angle_welded_to_doorjamb  ?a5)
                       (construction   ?a5 ?r))
                 (=>(steel_strike_plate_installed_to_doorjamb
           ?a6)
                       (construction   ?a6 ?r))
                 (=>(strike_plate_reinforcement_installed  ?a7)
                       (construction   ?a7 ?r)))
       (exists (?r1 ?r2 ?r3 ?r4 ?r5 ?r6 ?r7)
            (and (=>(doorjamb, (4) machine screws; (4) wood lags
?r1)
                     (material_of_construction  ?r1  ?a1))
                (=>(1 ¼"*8"h galvanized steel strut,(2) 5/6" X ¾"
nut & bolt?r2)
                     (material_of_construction  ?r2 ?a2))
                  (=>(two 5/16"x ¾" hex head nut and bolt screws
?r3)
                    (material_of_construction  ?r3 ?a3))
                (=>(three, 7 Gage, 10" x 1 ¼"" hinge
           reinforcement   ?r4)
                    (material_of_construction  ?r ?a4))
                  (=>(12 Gage galvanized steel angle   ?r5)
                     (material_of_construction  ?r ?a5))
                  (=>(4.86" x 1.75" x0.079" thick steel strike
           plate   ?r6)
                     (material_of_construction  ?r ?a6))
                  (=>(16 gage x 1 ¼"x6" strike plate reinforcement
           ?r7)
                     (material_of_construction  ?r ?a7))))))


(<=> (=>(and (metal door   ?x)
             (door leaf  ?r))
         (and (DWG_object ?x)
              (component  ?r ?x)))
   (exists (?a1 ?a2 ?a3 ?a4)
           (and (=>(door leaf hanged on door frame ?a1)
                        (construction   ?a1 ?r))
                  (=>(channel Welded to both faces sheets of door
leaf ?a2)
                        (construction   ?a2 ?r))
                   (=>(hinge reinforcement Installed at centre of
the leaf?a3)
                       (construction   ?a3 ?r))
                   (=>(lock reinforcement plate Welded to door
face sheet?a4)
                       (construction   ?a4 ?r)))
       (exists (?r1 ?r2 ?r3 ?4)
            (and (=>(35 ¾"w x 831/8"h x 1 3/4"d 20G galvanized
steel doorleaf ?r1)
                        (material_of_construction  ?r1 ?a1))
                    (=>(16 Gage flush top and bottom channel ?r2)
                        (material_of_construction  ?r2 ?a2)))
                   (=>((3) 9" x 1 ¼" x 7 Gage hinge
           reinforcement?r3)
                        (material_of_construction  ?r3 ?a3))
                  (=>(16 Gage steel lock reinforcement plate
           ?r4)
                        (material_of_construction  ?r4 ?a4))))))


(<=> (=>(and (metal door   ?x)
             (hardware & components      ?r))
         (and (DWG_object ?x)
              (component  ?r ?x)))
          (exists (?a1 ?a2 ?a3 ?a4)
                    (and (=>(lock fixed on door leaf  ?a1)
                             (construction   ?a1 ?r))
                          (=>(hinge installed in door leaf ?a2)
                            (construction   ?a2 ?r))
                           (=>(weatherstrip applied to door  ?a3)
                            (construction    ?a3 ?r)))
                (exists (?r1 ?r2 ?r3)
                    (and (=>((1) Yale 5307 Lever Lock  ?r1)
                              (material_of_construction    ?r1
?a1))
                            (=>(4 ¼" steel Hinges   ?r2)
                              (material_of_construction    ?r2
?a2)))
                           (=>(extruded aluminum & vinyl
           weatherstrip ?r3)
                              (material_of_construction    ?r3
?a3))))))
```

8.5.4 Architectural Design Data File in PSL Representations

The semantic translator (semantic translation definitions) for AutoCAD application takes/reads the architectural design data file containing AutoCAD application's terminology in PSL syntax (**section 8.3)** and produces a corresponding file containing only PSL terminology in PSL syntax (KIF) by substituting the definition of all the AutoCAD application's terminologies with definitions in PSL. In this translation, using the translation definitions for AutoCAD the architectural design data file in PSL syntax (KIF) is mapped to PSL definitions that contain only PSL terminology

```
(<=>(=>(metal_bldg-opening  ?x)
       (product   ?x))
    (exists (?r1  ?r2 ?r3)
       (and (=> (wood_frame ?r1)
                (resource    ?r1 ?x))
            (=> (8" with 3" steel_girt  ?r2)
                (resource    ?r2 ?x))
             (=> (wall_panel)
                 (resource    ?r3 ?x)))))


(<=> (=>(and (metal_bldg-opening ?x)
             (97"*102 ½" wood frame  ?r))
     (and (product ?x)
          (resource ?r ?x)))
       (exists (?a)
             (=>(and(wood-frame_constructed_as_design   ?a)
                    (processor_activity   ?a))
                    (creates   ?a  ?r))
       (exists (?r1)
               (=>(2*12 Douglas Fire Wood  ?r1)
                   (consumable  ?r1 ?a)))))


(<=> (=>(and (metal_bldg-opening ?x)
             (8" with 3" flange steel girt   ?r))
     (and (product ?x)
          (resource ?r ?x)))
     (exists (?a1 ?a2 )
             (=>(and (<=>(=>(clip attached to girt ?a1)
                            (processor_activity   ?a1))
                          (creates  ?a1  ?r))
                     (<=>(=> (clip attached to wooden frame ?a2)
                            (processor_activity   ?a2)))
                          (creates  ?a2  ?r)))
       (exists (?r1 ?r2 )
             (and (=>(girt, clip, six #12*1 ¼" Self Drilling
Screws ?r1)
                        (consumable   ?r1 ?a1))
                     (=>(four #12 *1 ½"Wood Lag Screws  ?r2)
                        (consumable     ?r2 ?a2)))))))
```

```
(<=> (=>(and (metal_bldg-opening ?x)
          (26 Gage 38"*108 ½"metal panel  ?r))
        (and (product ?x)
            (resource ?r ?x)))
     (exists (?a1 ?a2 )
          (=>(and (<=>(=>(wall panel attached to vertical
members ?a1)
                      (processor_activity   ?a1))
                      (creates  ?a1  ?r))
                 (<=>(=>(wall panel attached to door frame
?a2)
                      (processor_activity   ?a2))
                      (creates  ?a2  ?r)))
     (exists (?r1 ?r2 )
          (and (=>(metal panel,(22) #12*1 ¼" Self drilling
Screws ?r1)
                 (consumable  ?r1  ?a1 ))
               (=>(nine #12*1 ¼" Self drilling Screws?r2)
                 (consumable  ?r2  ?a2)))))))))


(<=>(=>(metal_door ?x)
      (product ?x))
      (exists (?r1  ?r2 ?r3)
        (and (=> (door_frame ?r1)
                (resource      ?r1 ?x))
             (=> (door_leaf  ?r2)
                 (resource   ?r2 ?x))
              (=>  (hard_ware_and_compenets ?r3)
                 (resource   ?r3 ?x)))))


(<=> (=>(and (metal_door ?x)
             (doorframe  ?r))
        (and (product ?x)
             (resource ?r ?x)))
  (exists (?a1 ?a2 ?a3 ?a4 ?a5 ?a6 ?a7)
  (=>(and (<=>(=>(jamb mounted to girt & wood frame ?a1)
                 (processor_activity      ?a1))
               (creates  ?a1  ?r))
          (<=>(=>(strut attached to horizontal girt and doorjamb ?a2)
                    (processor_activity   ?a2))
                  (creates  ?a2  ?r))
          (<=>(=> (jamb attached to frame head  ?a3)
                     (processor_activity     ?a3))
                  (creates  ?a3  ?r))
          (<=>(=> (Doorjamb prepared  ?a4)
                     (processor_activity      ?a4))
                  (creates  ?a  ?r))
          (<=>(=>(hinge reinforcement welded to doorjamb ?a4)
                     (processor_activtiy     ?a4))
                  (creates  ?a4  ?r))
          (<=>(=> (steel angle welded to doorjamb ?a5)
                     (processor_activtiy     ?a5))
                  (creates  ?a5  ?r))
          (<=>(=> (strike plate installed to doorjamb?a3)
                    (processor_activtiy     ?a6))
                  (creates  ?a6  ?r))
          (<=>(=> (strike plate reinforcement installed?a4)
                    (processor_activtiy     ?a7))
                  (creates  ?a7  ?r)))
(exists (?r1 ?r2 ?r3 ?r4 ?r5 ?r6 ?r7)
   (and (=>(doorjamb, (4)machine screws; (4)wood lags   ?r1)
           (consumable   ?r1 ?a1))
        (=>(1¼"*8"h galvanized steel strut,(2)5/6" X ¾" nut & bolt ?r2)
           (consumable  ?r2 ?a2))
        (=>(two 5/16" x ¾" hex head nut and bolt screws  ?r3)
          (consumable  ?r3 ?a3))
       (=>(three, 7 Gage, 10" x 1 ¼"" hinge reinforcement ?r4)
          (comsumable ?r4 ?a4))
       (=>(12 Gage galvanized steel angle ?r5)
          (comsumable  ?r5 ?a5)))
      (=>(4.86" x 1.75" x0.079" thick steel strike plate  ?r6)
         (comsumable?r6 ?a6)))
      (=>(16 gage x 1 ¼"x6" strike plate reinforcement  ?r7)
         (comsumable?r7 ?a7)))))))


(<=> (=>(and (metal_door ?x)
             (door leaf  ?r))
        (and (product ?x)
            (resource ?r ?x)))
     (exists (?a1 ?a2 ?a3 ?a4)
         (=>(and (<=>(=>(door leaf hanged on door frame ?a1)
                        (processor_activiy  ?a1))
                       (creates  ?a  ?r))
            (<=>(=> (channel Welded to faces sheets of door leaf
?a2)
                    (processor_activtiy  ?a2))
                    (creates  ?a  ?r))
            (<=>(=> (hinge reinforcement Installed at centre of
leaf?a3)
                       (processor_activtiy  ?a3))
                      (creates  ?a  ?r))
            (<=>(=> (lock reinforcement plate Welded to door
sheet ?a4)
                       (processor_activtiy  ?a4)))
                        (creates  ?a  ?r)))
         (exists (?r1 ?r2 ?r3 ?4)
             (and (=>(35 ¾"w x 831/8"h x 1 3/4"d 20G galvanized
steel doorleaf ?r1)
                       (comsumable   ?r1 ?a1))
                    (=>(16 Gage flush top and bottom channel ?r2)
                       (comsumable    ?r2 ?a2)))
                    (=>((3) 9" x 1 ¼" x 7 Gage hinge
           reinforcement?r3)
                       (comsumable    ?r3 ?a3))
                   (=>(16 Gage steel lock reinforcement plate
           ?r4)
                       (comsumable  ?r4 ?a4)))))))

(<=> (=>(and (metal_door ?x)
             (hardware & components   ?r))
        (and (product ?x)
            (resource ?r ?x)))
     (exists (?a1 ?a2 ?a3 ?a4)
             (=>(and (<=>(=>(lock fixed on door leaf  ?a1)
                            (processor_activtiy     ?a1 ?a))
                      (creates  ?a  ?r))
                    (<=>(=> (hinge installed in door leaf ?a2)
                            (processor_activtiy     ?a2 ?a))
                      (creates  ?a  ?r))
                     (<=>(=> (weatherstrip applied to door  ?a3)
                            (processor_activtiy     ?a3 ?a))
                       (creates  ?a  ?r)))
       (exists (?r1 ?r2 ?r3)
              (and (=>((1) Yale 5307 Lever Lock  ?r1)
                       (comsumable    ?r1 ?a1))
                    (=>(4 ¼" steel Hinges   ?r2)
                       (comsumable    ?r2 ?a2)))
                 (=>(extruded aluminum & vinyl weatherstrip ?r3)
                       (comsumable   ?r3 ?a3)))))))
```

8.5.5 Architectural Design Data File in PSL syntax and CCS terminology

In this translation a reverse step is followed to translate the AutoCAD application's architectural design data file containing only PSL terminology in PSL(KIF notation) syntax (**section 9.5.4**) into CCS application's terminology but still preserving the PSL (KIF notation) syntax intact. Using the semantic translator (semantic translation definitions) for CCS the AutoCAD's architectural design data file in PSL representations (PSL syntax and terminology) is mapped to a file containing only CCS application's terminology to produce architectural design data file containing only CCS application's terminology in PSL (KIF notation) syntax as input data for CCS cost estimating application.

```
(<=>(=>(metal_bldg-opening  ?x)
      (bill_of_quantity_item    ?x))
    (exists (?r1  ?r2 ?r3)
      (and (=> (97"*102 ½" wood_frame ?r1)
               (macro_op       ?r1   ?x))
           (=> (8" with 3" steel_girt  ?r2)
               (macro_op       ?r2   ?x))
            (=> (wall_panel)
                (macro_op  ?r3   ?x)))))
```

```
(<=> (=>(and (metal_bldg-opening ?x)
             (97"*102 ½" wood frame  ?r))
        (and (bill_of_quantity_item     ?x)
             (macro_op   ?r ?x)))
        (exists (?a)
             (=>(wood-frame_constructed_as_design   ?a)
                (operation   ?a ?r))
       (exists (?r1)
              (=>(2*12 Douglas Fire Wood  ?r1)
                 (material    ?r1 ?a)))))
```

```
(<=> (=>(and (metal_bldg-opening ?x)
             (8" with 3" flange steel girt   ?r))
        (and (bill_of_quantity_item     ?x)
             (macro_op   ?r ?x)))
     (exists (?a1 ?a2 )
             (=>(and (=>(clip attached to girt ?a1)
                        (operation  ?a1 ?r))
                    (=> (clip attached to wooden frame ?a2)
                        (operation  ?a2 ?r)))
      (exists (?r1 ?r2 )
             (and (=>(girt, clip, six #12*1 ¼" Self Drilling
Screws ?r1)
                       (material   ?r1 ?a1))
                    (=>(four #12 *1 ½"Wood Lag Screws  ?r2)
                       (material     ?r2 ?a2)))))))
```

```
(<=> (=>(and (metal_bldg-opening ?x)
             (26 Gage 38"*108 ½"metal panel  ?r))
        (and (bill_of_quantity_item     ?x)
             (macro_op   ?r ?x)))
      (exists (?a1 ?a2 )
             (=>(and (=>(wall panel attached to vertical members
?a1)
                            (operation  ?a1 ?r))
                     (=>(wall panel attached to door frame ?a2)
                            (operation   ?a2 ?r)))
         (exists (?r1 ?r2 )
              (and (=>(metal panel,(22) #12*1 ¼" Self drilling
Screws ?r1)
                      (material  ?r1  ?a1 ))
                   (=>(nine #12*1 ¼" Self drilling Screws?r2)
                      (material  ?r2  ?a2)))))))


(forall (?r)
       (=>(metal_door ?x)
           (bill_of_quantity_item      ?x))
     (exists (?r1  ?r2 ?r3)
       (and (=> (door_frame ?r1)
                (macro_op      ?r1 ?x))
             (=> (door_leaf  ?r2)
                 (macro_op      ?r2 ?x))
             (=> (hard_ware_and_compenets ?r3)
                  (macro_op      ?r3 ?x)))))


(<=> (=>(doorframe  ?r)
        (macro_op   ?r))
      (exists (?a1 ?a2 ?a3 ?a4 ?a5 ?a6 ?a7)
         (=>(and (=>(jamb mounted to girt & wood frame ?a1)
                    (operation  ?a1 ?r))
                 (=>(strut attached to horizontal girt and
doorjamb ?a2)
                    (operation  ?a2 ?r))
                   (=> (jamb attached to frame head  ?a3)
                     (operation   ?a3 ?r))
                     (creates  ?a3  ?r))
                   (=> (Doorjamb prepared  ?a4)
                     (operation    ?a4 ?r))
                 (=>(hinge reinforcement welded to doorjamb ?a4)
                      (operation   ?a4  ?r))
                  (=> (steel angle welded to doorjamb ?a5)
                      (operation  ?a5  ?r))
                   (=> (strike plate installed to doorjamb?a3)
                     (operation   ?a6 ?r))
                   (=> (strike plate reinforcement installed?a4)
                     (operation  ?a7  ?r)))
      (exists (?r1 ?r2 ?r3 ?r4 ?r5 ?r6 ?r7)
           (and (=>(doorjamb, (4)machine screws; (4)wood lags
?r1)
                        (material   ?r1 ?a1))
                (=>(1¼"*8"h galvanized steel strut,(2) nut & bolt
?r2)
                        (material  ?r2 ?a2))
                (=>(two 5/16" x ¾" hex head nut and bolt screws
            ?r3)
                        (material  ?r3 ?a3))
         (=>(three, 7 Gage, 10" x 1 ¼"" hinge reinforcement ?r4)
                       (material ?r4 ?a4))
              (=>(12 Gage galvanized steel angle ?r5)
                       (material  ?r5 ?a5)))
            (=>(4.86" x 1.75" x0.079" thick steel strike plate
           ?r6)
              (material  ?r6 ?a6)))
           (=>(16 gage x 1 ¼"x6" strike plate reinforcement  ?r7)
              (material ?r7 ?a7)))))))
```

```
(<=> (=>(door leaf  ?r)
         (macro_op   ?r))
    (exists (?a1 ?a2 ?a3 ?a4)
         (=>(and(=>(door leaf hanged on door frame ?a1)
                    (operation   ?a1 ?r))
           (=> (channel Welded to both faces sheets of door leaf
?a2)
               (operation   ?a2  ?r))
            (=> (hinge reinforcement Installed at centre of the
leaf?a3)
               (operation  ?a3  ?r))
             (=>  (lock reinforcement plate Welded to door face
sheet?a4)
                (operation  ?a4 ?r)))
(exists (?r1 ?r2 ?r3 ?4)
      (and (=>(35¾"wx831/8"hx13/4"d 20G galvanized steel doorleaf
?r1)
                  (material   ?r1 ?a1))
             (=>(16 Gage flush top and bottom channel ?r2)
                (material    ?r2 ?a2)))
             (=>((3) 9" x 1 ¼" x 7 Gage hinge reinforcement?r3)
                (material   ?r3 ?a3))
             (=>(16 Gage steel lock reinforcement plate ?r4)
                (material  ?r4 ?a4)))))))
```

```
(<=> (=>(hardware & components   ?r))
        (macro_op   ?r))
      (exists (?a1 ?a2 ?a3 ?a4)
             (=>(and (=>(lock fixed on door leaf  ?a1)
                            (operation   ?a1 ?r))
                     (=> (hinge installed in door leaf ?a2)
                             (operation  ?a2 ?r))
                      (=> (weatherstrip applied to door  ?a3)
                            (operation  ?a3 ?r)))
       (exists (?r1 ?r2 ?r3)
            (and (=>((1) Yale 5307 Lever Lock  ?r1)
                        (material    ?r1 ?a1))
                     (=>(4 ¼" steel Hinges   ?r2)
                        (material   ?r2 ?a2)))
                    (=>(extruded aluminum & vinyl weatherstrip
           ?r3)
                      (material   ?r3 ?a3)))))))
```

8.5.6 Architectural Design Data File in CCS Application's input File

In this translation the syntactic translator maps the architectural design data file containing only CCS application's terminology in PSL (KIF notation) syntax. **(Section 9.5.5)** back into CCS application's syntax and terminology The result is the CCS application's representation of the architectural design data as cost estimating input data This contains only the architectural design data as input for cost estimating application but this with other cost estimating data represents the CCS application's native ontology

```
(project  metal door in a metal building opening)

[bill of quantities]
(Item Number                    )
(Op Code                        )
(bill o quantity item           metal building-opening)
(bill o quantity item           metal door)
(Op Code Unit                   )
(Billed Quantity                )
(Nett Rate                      )
(Nett Amount                    )


[macro for Metal Building Opening]
(Item Number                 )
(Op Code                     )
(Macro Op Code Description    97"*102 ½" wood frame)
(Macro Op Code Description    steel girt)
(Macro Op Code Description    metal panel)
(Op Code Unit                )
(Billed Quantity             )
(Nett Rate                   )
(Nett Amount                 )


[operations for 97"*102 ½" wood frame]
(Item Number                 )
(Op Code                     )
(Op Code Description            wood-frame construction as design)
(Op Code Unit                )
(Billed Quantity             )
(Nett Rate                   )
(Nett Amount                 )
[worksheet]
(resource code               )
(labour                      )
(plant                       )
(material           2*12 Douglas Fire Wood)
(rates                       )

[operations for steel girt]
(Item Number                 )
(Op Code                     )
(Op Code Description         clip attached to girt)
(Op Code Description         clip attached to wooden frame)
(Op Code Unit                )
(Billed Quantity             )
(Nett Rate                   )
(Nett Amount                 )
[worksheet]
(resource code               )
```

```
(labour                 )
(plant                  )
(material          girt, clip, six #12*1 ¼" Self Drilling Screws)
(material          four #12 *1 ½"Wood Lag Screws)
(rates                  )

[operations for 26 Gage 38"*108 ½"metal panel]
(Item Number            )
(Op Code                )
(Op Code Description    wall panel attached to vertical members)
(Op Code Description    wall panel attached to doorframe)
(Op Code Unit           )
(Billed Quantity        )
(Nett Rate              )
(Nett Amount            )
[worksheet]
(resource code          )
(labour                 )
(plant                  )
(material          metal panel,(22) #12*1 ¼" Self drilling Screws)
(material          nine #12*1 ¼" Self drilling Screws)
(rates                  )


[macro for Metal Door]
(Item Number            )
(Op Code                )
(Macro Op Code Description    doorframe constructed as design  )
(Macro Op Code Description    doorleaf constructed as design and
hanged)
(Macro Op Code Description    hard ware and components fixed on door)
(Op Code Unit           )
(Billed Quantity        )
(Nett Rate              )
(Nett Amount            )


[operations for doorframe]
(Item Number            )
(Op Code                )
(Op Code Description          jamb mounted to girt & wood frame)
(Op Code Description          strut attached to horizontal girt and
doorjamb)
(Op Code Description          jamb attached to frame head  )
(Macro Op Code Description    Doorjamb prepared for connetion with
other parts)
(Op Code Description          hinge reinforcement welded to doorjamb)
(Op Code Description          steel angle welded to doorjamb)
(Op Code Description          strike plate installed to doorjamb)
(Op Code Description                  strike plate reinforcement
installed)
(Op Code Unit           )
(Billed Quantity        )
(Nett Rate              )
(Nett Amount            )
[worksheet]
(resource code          )
(labour                 )
(plant                  )
(material          doorjamb, (4)machine screws; (4)wood lags   )
(material          1¼"*8"h galvanized steel strut,(2)5/6" X ¾" nut &
bolt)
(material          two 5/16" x ¾" hex head nut and bolt screws  )
(material          three, 7 Gage, 10" x 1 ¼"" hinge reinforcement)
(material          12 Gage galvanized steel angle)
(material          4.86" x 1.75" x0.079" thick steel strike plate)
(material          16 gage x 1 ¼"x6" strike plate reinforcement)
(rates                  )
```

```
[operations for door leaf]
(Item Number            )
(Op Code                )
(Op Code Description          door leaf hanged on door frame)
(Op Code Description          channel Welded to both faces sheets of
door leaf)
(Op Code Description          hinge reinforcement Installed at centre
of the leaf)
(Op Code Description          lock reinforcement plate Welded to door
face sheet)
 (Op Code Unit          )
(Billed Quantity        )
(Nett Rate              )
(Nett Amount            )
[worksheet]
(resource code          )
(labour                 )
(plant                  )
(material        35 ¾"w x 831/8"h x 1 3/4"d 20G galvanized steel
doorleaf)
(material        16 Gage flush top and bottom channel)
(material        (3) 9" x 1 ¼" x 7 Gage hinge reinforcement)
(material        16 Gage steel lock reinforcement plate)
(rates                  )

[operations for hard ware and components]
(Item Number            )
(Op Code                )
(Op Code Description          lock fixed on door leaf  )
(Op Code Description          hinge installed in door leaf)
(Op Code Description          weatherstrip applied to door )
(Op Code Unit           )
(Billed Quantity        )
(Nett Rate              )
(Nett Amount            )
[worksheet]
(resource code          )
(labour                 )
(plant                  )
(material        (1) Yale 5307 Lever Lock   )
(material        4 ¼" steel Hinges   )
(material        extruded aluminum & vinyl weatherstrip)
(rates                  )
```

8.5.7 Cost Estimating Output File in CCS Application's Representations

This represents the output file of CCS cost estimating application that is generated based on the input file (shown in **section 9.5.6)** translated from AutoCAD data representation to CCS using PSL.

```
[BILLED QUNATITIES NETT AMOUNT]

(Item Number                                 )
(Op Code                                     )
(Op Code Description          construct wood frame)
(Op Code Unit                                )
(Billed Quantity                             )
(AMOUNT                       1000)----------
(LABOUR                       500)---&
(PLANT                        250)----&
(MATERIAL                     250)-----
```

```
(TEMP.MATERIAL                )
(OVERHEADS                    )
(PROV.SUMS                    )
(SUBCONSTRACT                 )
(MARKUP                       )

[ BILL RESOURCE ANALYSIS for all trades]

(RESOURCE CODE                )
(RESOURCE NAMES               2*12 Douglas Fire Wood)
(UNIT                         )
(GROUP                        )
(RATE                         x)
(UTILIZATION                  y)
(AMOUNT                       250)
(LABOUR                       )
(PLANT                        )
(MATERIAL                     250)
(TEMP.MATERIAL                )
(OVERHEADS                    )
(PROV.SUMS                    )
(SUBCONSTRACT                 )

[RESOURCE by BILL ITEM]

 (ITEM                              )
 (OP CODE                           )
 (OP CODE DESCRIPTION               construct wood frame)
 ( QTY                              )
 ( UNIT                             )
 ( RATE =                           )
RESOURCES INFORMATION
   (RESOURCE GROUP                  )
   (RESOURCE CODE                   )
   (RESOURCE NAME                   2*12 Douglas Fire Wood)
   (CURRANCY                        )
   (QUANTITY                        )
   (RATE                            )
   (AMOUNT                          250)
   (PRODUCTION                      )
```

8.5.8 Cost Estimating data in PSL syntax and CCS terminology

The syntactic translator reads the cost estimating output file (**section 9.5.7**) and produces a corresponding file using PSL syntax, but still persevering the CCS application's terminology intact. This file is the same as the file in **section 9.5.5** but includes other and CCS specific cost estimating data such as resource, labour, plant, cost etc

```
(<=>(=>(metal_bldg-opening  ?x)
       (bill_item      ?x))
      (exists (?N)
                (=> (1000  ?N)
                    (amount   ?N ?x))))


(<=> (=>(construct wood frame  ?r)
        (bill_item      ?r))
        (exists (?N)
                (=> (1000  ?N)
                   (amount   ?N ?x))))
```

```
(<=> (=>(attach steel girt to wood frame   ?r)
       (bill_item      ?r))
      (exists (?N)
             (=> (1000  ?N)
                (amount   ?N ?x))))

  (<=> (=>(install metal panel  ?r)
       (bill_item      ?r))
      (exists (?N)
             (=> (1000  ?N)
                (amount   ?N ?x))))


(<=> (=>(2*12 Douglas Fire Wood  ?r)
       (resource name      ?r))
      (exists (?N)
             (=> (250 ?N)
                (amount   ?N ?x))))


(and (=>(girt, clip, six #12*1 ¼" Self Drilling Screws  ?r)
        (resource name      ?r))
     (exists (?N)
        (=> (250  ?N)
            (amount   ?N ?x))))


(<=>(=>(metal panel,(22) #12*1 ¼" Self drilling Screws  ?r)
        (resource name      ?r))
     (exists (?N)
       (=> (250  ?N)
           (amount   ?N ?x))))

(<=> (=>(construct wood frame  ?a)
       (bill item  ?a))
     (exists (?r1 ?r2 ?r3)
             (=>(and (=>(2*12 Douglas Fire Wood  ?r1)
                       (resource name     ?r1 ?a))
                  (=>(carpenter  ?r2)
                       (resource name     ?r2 ?a))
                    (=>(machine   ?r3)
                       (resource name     ?r3 ?a)))
          (exists (?N1 ?N2 ?N3)
               (and (=>(250  ?N1)
                       (amount   ?N1   ?r1))
                    (=>(200  ?r2)
                       (amount   ?N2 ?r2))
                      (=>(300    ?r3)
                       (amount   ?N3 ?r3)))))))


(<=> (=>(attach girt to wood frame   ?a)
       (billitem        ?a))
    (exists (?r1 ?r2 ?r3 ?r4)
      (=>(and (=>(girt, clip, six #12*1 ¼" Self Drilling Screws
?r1)
                   (resource name     ?r1 ?a))
                 (=>(four #12 *1 ½"Wood Lag Screws  ?r2)
                       (resource name    ?r2 ?a))
                  (=>(carpenter ?r3)
                       (resource name   ?r3  ?a))
                   (=>(machine   ?r4)

                      (resource name   ?r4  ?a)))
       (exists (?N1 ?N2 ?N3)
               (and (=>(250  ?N1)
                       (amount   ?N1   ?r1))
                   (=>(200  ?r2)
                       (amount   ?N2 ?r2))
                     (=>(300    ?r3)
                       (amount   ?N3 ?r3))
                      (=>(400    ?r4)
                       (amount   ?N4 ?r3)))))))
```

```
(<=> (=>(attach metal panel to vertical member & doorframe  ?a)
       (bill item      ?a))
  (exists (?r1 ?r2 ?r3 ?r4)
       (=>(and (=>(metal panel,(22) #12*1 ¼" Self drilling
Screws ?r1)
               (material  ?r1   ?a1 ))
            (=>(nine #12*1 ¼" Self drilling Screws?r2)
(material  ?r2  ?a2))
             (=>(carpenter  ?r2)
                (labour    ?r2   ?a1 ?a2))
             (=>(machine   ?r3)
                (plant     ?r3   ?a1 ?a2)))
  (exists (?N1 ?N2 ?N3)
               (and (=>(250  ?N1)
                       (amount   ?N1   ?r1))
                   (=>(200  ?r2)
                       (amount   ?N2 ?r2))
                     (=>(300    ?r3)
                       (amount   ?N3 ?r3))
                      (=>(400    ?r4)
                         (amount   ?N4 ?r3)))))))
```

8.5.9 CCS Cost Estimating data in PSL Representations

The semantic translator (semantic translation definitions) for CCS application reads the cost-estimate data file written in the application's terminology and PSL (KIF notation) syntax (**section 9.5.8**) and produces a corresponding file containing only PSL terminology in PSL syntax (KIF) by substituting the definition of all the CCS application's terminologies in **section 9.5.8** with definitions in PSL. Using the translation definitions for CCS the cost-estimate data file in PSL syntax (KIF) and the application's terminology of **section 9.5.8** is mapped to PSL definitions that contain only PSL terminology.

```
(<=>(=>(metal_bldg-opening  ?a)
      (activity      ?a))
     (exists (?N)
       (=> (1000  ?N)
           (cost of activity   ?N ?a)))))
```

```
(<=> (=>(construct wood frame  ?a)
       (activity     ?a))
       (exists (?N)
              (=> (1000  ?N)
                 (cost of activity   ?N ?a))))


(<=> (=>(attach steel girt to wood frame   ?a)
       (activity     ?a))
       (exists (?N)
              (=> (1000  ?N)
                 (cost of activity   ?N ?a))))

  (<=> (=>(install metal panel  ?a)
       (activity     ?a))
       (exists (?N)
              (=> (1000  ?N)
                 (cost of activity   ?N ?a))))
(<=> (=>(2*12 Douglas Fire Wood  ?r)
       (consumable     ?r))
       (exists (?N)
              (=> (250  ?N)
                 (cost of resource  ?N ?r))))


(and (=>(girt, clip, six #12*1 ¼" Self Drilling Screws  ?r)
        (consumable      ?r))
     (exists (?N)
        (=> (250  ?N)
            (cost of resource  ?N ?r))))


(<=>(=>(metal panel,(22) #12*1 ¼" Self drilling Screws  ?r)
        (consumable   ?r))
     (exists (?N)
       (=> (250  ?N)
           (cost of resource  ?N ?r))))


(<=> (=>(construct wood frame  ?a)
       (activity   ?a))
     (exists (?r1 ?r2 ?r3)
              (=>(and (=>(2*12 Douglas Fire Wood  ?r1)
                     (consumable      ?r1 ?a))
                  (=>(carpenter  ?r2)
                     (reusable     ?r2 ?a))
                   (=>(machine   ?r3)
                     (reusable     ?r3 ?a)))
           (exists (?N1 ?N2 ?N3)
                (and (=>(250  ?N1)
                       (cost of resource  ?N1 ?r1))
                    (=>(200  ?r2)
                       (cost of resource  ?N2 ?r2))
                     (=>(300    ?r3)
                       (cost of resource  ?N3 ?r3)))))))
```

```
(<=> (=>(attach girt to wood frame   ?a)
        (activity        ?a))
    (exists (?r1 ?r2 ?r3 ?r4)
       (=>(and (=>(girt, clip, six #12*1 ¼" Self Drilling Screws
?r1)
                 (consumable   ?r1 ?a))
               (=>(four #12 *1 ¼"Wood Lag Screws  ?r2)
                  (consumable   ?r2 ?a))
                (=>(carpenter  ?r3)
                     (reusable    ?r3  ?a))
                 (=>(machine   ?r4)
                    (reusable   ?r4  ?a)))
       (exists (?N1 ?N2 ?N3)

                (and (=>(250  ?N1)
                       (cost of resource    ?N1   ?r1))
                    (=>(200  ?r2)
                       (cost of resource    ?N2 ?r2))
                     (=>(300    ?r3)
                       (cost of resource  ?N3 ?r3))
                      (=>(400    ?r4)
                       (cost of resource    ?N4 ?r3)))))))


(<=> (=>(attach metal panel to vertical member & doorframe  ?a)
        (activity      ?a))
  (exists (?r1 ?r2 ?r3 ?r4)
        (=>(and (=>(metal panel,(22) #12*1 ¼" Self drilling
Screws ?r1)
               (consumable  ?r1  ?a1 ))
            (=>(nine #12*1 ¼" Self drilling Screws?r2)
               (consumable  ?r2  ?a2))
             (=>(carpenter  ?r2)
                (reusable    ?r2   ?a1 ?a2))
             (=>(machine   ?r3)
                (reusable    ?r3   ?a1 ?a2)))
  (exists (?N1 ?N2 ?N3)
                (and (=>(250  ?N1)
                       (cost of resource    ?N1   ?r1))
                    (=>(200  ?r2)
                       (cost of resource    ?N2 ?r2))
                     (=>(300    ?r3)
                       (cost of resource  ?N3 ?r3))
                      (=>(400    ?r4)
                         (cost of resource    ?N4 ?r3)))))))
```

8.5.10 Cost-Estimating data in PSL syntax and MS project terminology

In this translation a reverse step is followed to translate the CCS application's Cost estimate data file containing only PSL terminology in PSL (KIF notation) syntax (**section 9.5.9**) into a file containing only Microsoft project application's terminology but sill preserving the PSL syntax intact. Using the semantic translator (semantic translation definitions) for Microsoft project Scheduling application the CCS cost estimating data file in PSL representation is mapped to a file containing only Microsoft project

Schedulıng applıcatıon’s termınologıes to produce cost estımatıng data file wrıtten usıng Microsoft proȷect applıcatıon’s termınology and PSL syntax as ınput data fire for the schedulıng applıcatıon.

```
(<=>(=>(metal_bldg-opening  ?a)
      (summary task      ?a))
     (exısts (?N)
       (=> (1000  ?N)
           (total cost  ?N ?a))))


(<=>  (=>(construct wood frame   ?a)
        (task      ?a))
       (exısts (?N)
              (=> (1000  ?N)
                 (total cost   ?N ?a))))

(<=>  (=>(attach steel gırt to wood frame   ?a)
        (summary task      ?a))
       (exısts (?N)
              (=> (1000  ?N)
                 (total task cost   ?N ?a))))

  (<=>  (=>(ınstall metal panel  ?a)
        (summary task      ?a))
       (exısts (?N)
              (=> (1000  ?N)
                 (total cost   ?N ?a))))


(<=>  (=>(2*12 Douglas Fıre Wood  ?r)
        (resource name      ?r))
       (exısts (?N)
              (=> (250  ?N)
                 (cost/use  ?N ?r))))

(and (=>(gırt, clıp, six #12*1 ¼" Self Drıllıng Screws  ?r)
         (resource name               ?r))
     (exısts (?N)
        (=> (250  ?N)
            (cost/use  ?N ?r))))

(<=>(=>(metal panel,(22) #12*1 ¼" Self drıllıng Screws  ?r)
        (resource name      ?r))
     (exısts (?N)
       (=> (250  ?N)
           (cost/use  ?N ?r))))

(<=>  (=>(construct wood frame  ?a)
        (task   ?a))
     (exists (?r1 ?r2 ?r3)
             (=>(and (=>(2*12 Douglas Fıre Wood  ?r1)
                    (resource name     ?r1 ?a))
                 (=>(carpenter  ?r2)
                    (resource name     ?r2 ?a))
                 (=>(machine   ?r3)
                    (resource name     ?r3 ?a)))
           (exısts (?N1 ?N2 ?N3)
               (and (=>(250  ?N1)
                      (cost/use  ?N1 ?r1))
                   (=>(200  ?r2)
                      (cost/use  ?N2 ?r2))
                   (=>(300   ?r3)
                      (cost/use  ?N3 ?r3)))))))
```

```
(<=> (=>(attach girt to wood frame   ?a)
        (summary task        ?a))
    (exists (?r1 ?r2 ?r3 ?r4)
       (=>(and (=>(girt, clip, six #12*1 ¼" Self Drilling Screws ?r1)
                  (resource name   ?r1 ?a))
                (=>(four #12 *1 ½"Wood Lag Screws  ?r2)
                   (resource name   ?r2 ?a))
                 (=>(carpenter  ?r3)
                      (resource name    ?r3  ?a))
                 (=>(machine   ?r4)
                    (resource name   ?r4  ?a)))
       (exists (?N1 ?N2 ?N3)
                (and (=>(250  ?N1)
                         (cost/use    ?N1   ?r1))
                     (=>(200  ?r2)
                         (cost/use    ?N2 ?r2))
                     (=>(300    ?r3)
                         (cost/use  ?N3 ?r3))
                      (=>(400    ?r4)
                          (cost/use    ?N4 ?r3)))))))
```

```
(<=> (=>(attach metal panel to vertical member & doorframe  ?a)
        (summary task      ?a))
  (exists (?r1 ?r2 ?r3 ?r4)
        (=>(and (=>(metal panel,(22) #12*1 ¼" Self drilling
Screws ?r1)
                (resource name  ?r1  ?a1 ))
            (=>(nine #12*1 ¼" Self drilling Screws?r2)
                (resource name  ?r2  ?a2))
             (=>(carpenter  ?r2)
                 (resource name    ?r2   ?a1 ?a2))
              (=>(machine   ?r3)
                 (resource name    ?r3   ?a1 ?a2)))
  (exists (?N1 ?N2 ?N3)
               (and (=>(250  ?N1)
                        (cost/use    ?N1   ?r1))
                    (=>(200  ?r2)
                        (cost/use    ?N2 ?r2))
                      (=>(300    ?r3)
                        (cost/use  ?N3 ?r3))
                       (=>(400    ?r4)
                       (cost/use    ?N4 ?r3)))))))
```

8.5.11 Cost-Estimating data in MS Project Application's Representations

In this translation the syntactic translator maps the file in section **9.5.10** i.e. the CCS cost-estimate data file in PSL syntax (KIF notation) and Microsoft project scheduling application's terminology back into Microsoft project Scheduling application's representation (Microsoft project syntax and terminology). The result is the Microsoft project Scheduling application's representation of the cost-estimate data as scheduling input data including scheduling specific data related to CCS cost estimating and AutoCAD design application.

```
GANTT CHART ENTRY TASK INFORMATION
(ID             )
(WBS            )
(CONSTRAINTS    )
(TASK NAME           construct wood)
(DURATION       )
(START TIME          )
( FINISH TIME        )
(PREDECESSORS        )
(FIXED COST          £00)
(TOTAL COST          £100)
(RESOURCE NAMES    2*12 Douglas Fire Wood, carpenter,
                  carpentery_equipment)
(COST/USE            £100)

GANTT CHART ENTRY TASK INFORMATION
(ID        )
(WBS       )
(CONSTRAINTS    )
(TASK NAME SUMMARY   attach steel gırt to wood frame)
(DURATION       )
(START TIME          )
( FINISH TIME        )
(PREDECESSORS        )
(FIXED COST          £00)
(TOTAL COST          £100)
(RESOURCE NAMES      )
(COST/USE            £100)


GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY    attaché  steel girt to wood frame)
(ID        )
(WBS       )
(CONSTRAINTS    )
(TASK NAME                 attach clıp to gırt)
(DURATION       )
(START TIME          )
( FINISH TIME        )
(PREDECESSORS   )
(FIXED COST          £00)
(TOTAL COST          £100)
(RESOURCE NAMES    gırt, clip, six #12*1 ¼" Self Drilling Screws,
                     carpenter, carpentry equipment)
(COST/USE            £100)


GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY    attach  steel girt to wood frame)
(ID        )
(WBS       )
(CONSTRAINTS    )
(TASK NAME           attach clıp to wooden frame)
(DURATION       )
(START TIME          )
( FINISH TIME        )
(PREDECESSORS   )
(FIXED COST          £00)
(TOTAL COST          £100)
(RESOURCE NAMES   four#12*1½"Wood Lag Screws, carpenter,
                  carpentry equipment)
(COST/USE            £00)
```

GANTT CHART ENTRY TASK INFORMATION

```
(ID        )
(WBS       )
(CONSTRAINTS     )
(TASK NAME SUMMAY     attach metal panel to vertical members &
                       doorframe)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       )
(COST/USE             £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(TASK NAME SUMMAY     attach metal wall panel to vertical
                       members & door frame)
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME            attach wall panel to vertical members)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       metal panel,(22) #12*1 ¼" Self drilling
                       Screws, carpenter, carpentery_equipment)
(COST/USE             £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(TASK NAME SUMMAY     attach  metal panel to vertical members &
                       door frame)
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME            attach panel to doorframe)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       nine #12*1 ¼" Self drilling Screws,
                       carpenter, carpentery_equipment)
(COST/USE             £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(ID              )
(WBS             )
(CONSTRAINTS     )
(TASK NAME SUMMAY     construct doorframe)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       )
(COST/USE             £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(TASK NAME SUMMAY     construct doorframe)
(ID              )
(WBS             )
(CONSTRAINTS     )
(TASK NAME       mount jamb to girt & wood frame)
(DURATION        )
(START TIME           )
(FINISH TIME          )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       doorjamb, (4)machine screws; (4)wood
                       lags, carpenter, carpentery_equipment)
(COST/USE             £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(TASK NAME SUMMAY     construct doorframe)
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME            attach   strut   to   horizontal   girt   and
doorjamb)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES   1¼"*8"h galvanized steel strut,(2)5/6" X
                 ¾" nut & bolt, carpenter,carpentery_equipment)
(COST/USE           £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(TASK NAME SUMMAY     construct doorframe)
(ID              )
(WBS             )
(CONSTRAINTS          )
(TASK NAME       attach jamb to frame head  )
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       two 5/16" x ¾" hex head nut and bolt
                  screws  , carpenter, carpentery_equipment)
(COST/USE        £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME SUMMARY    prepare Doorjamb )
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       )

(COST/USE             £100)
```

```
GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY     prepar Doorjamb)
(ID         )
(WBS        )
(CONSTRAINTS     )
(TASK NAME            weld hinge reinforcement to doorjamb)
(DURATION        )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS    )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES     three, 7 Gage, 10" x 1 ¼"" hinge
                   reinforcement, carpenter,
                   carpentery_equipment)
(COST/USE         £100)

GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY     prepare Doorjamb)
(ID              )
(WBS             )
(CONSTRAINTS     )
(TASK NAME       weld steel angle to doorjamb)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)

(RESOURCE NAMES       12 Gage galvanized steel angle,carpenter,
                       carpentery_equipment)
(COST/USE        £100)

GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY     prepare Doorjamb)
(ID         )
(WBS        )
(CONSTRAINTS     )
(TASK NAME       install strike plate to doorjamb)
(DURATION        )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS    )
(FIXED COST           £00)
(TOTAL COST           £100)

(RESOURCE NAMES        4.86" x 1.75" x0.079" thick steel strike
                       plate,carpenter, carpentery_equipment)

(COST/USE             £100)


GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY     prepare Doorjamb)
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME       install strike plate reinforcement)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
```

```
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       16 gage x 1 ¼"x6" strike plate
                      reinforcement?,carpenter,
                      carpentery_equipment)
(COST/USE             £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME SUMMARY       construct door leaf)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       )
(COST/USE                 £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(TASK NAME SUMMAY     construct door leaf)
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME            hang door leaf on door frame)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       35 ¾"w x 831/8"h x 1 3/4"d 20G galvanized
                       steel doorleaf, carpenter,
                       carpentery_equipment)
(COST/USE             £100)
```

GANTT CHART ENTRY TASK INFORMATION

```
(TASK NAME SUMMAY     construct door leaf)
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME        Weld channel to both faces sheets of door leaf)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES      16 Gage flush top and bottom channel,
                       carpenter, carpentery_equipment)
(COST/USE            £100
```

GANTT CHART ENTRY TASK INFORMATION

```
(TASK NAME SUMMAY     construct door leaf)
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME            Install  hinge  reinforcement  at  centre  of
the leaf)
```

```
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES        (3) 9" x 1 ¼" x 7 Gage hinge
                       reinforcement, carpenter,
                       carpentery_equipment)
(COST/USE             £100)


GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY     construct door leaf)
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME      Weld lock reinforcement plate to door face
                 sheet)
(DURATION             )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES    16 Gage steel lock reinforcement plate,
                       carpenter, carpentery_equipment)
(COST/USE        £100)


GANTT CHART ENTRY TASK INFORMATION
(ID                   )
(WBS                  )
(CONSTRAINTS          )
(TASK NAME SUMMARY    fix lock on door leaf)
(DURATION             )
(START TIME           )
( FINISH TIME      .  )
(PREDECESSORS         )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES       )
(COST/USE             £100)


GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY     fix hardware & components to door)
(ID          )
(WBS         )
(CONSTRAINTS     )
(TASK NAME            fix lock on door leaf)
(DURATION        )
(START TIME           )
( FINISH TIME         )
(PREDECESSORS    )
(FIXED COST           £00)
(TOTAL COST           £100)
(RESOURCE NAMES        (1) Yale 5307 Lever Lock, carpenter,
                       carpentery_equipment)
(COST/USE             £100)
```

```
GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY      fix hardware & components to door)
(ID                    )
(WBS                   )
(CONSTRAINTS           )
(TASK NAME             install hinge in door leaf)
(DURATION              )
(START TIME            )
( FINISH TIME          )
(PREDECESSORS          )
(FIXED COST            £00)
(TOTAL COST            £100)
(RESOURCE NAMES         4 ¼" steel Hinges, carpenter,
                        carpentery_equipment)
(COST/USE              £100)

GANTT CHART ENTRY TASK INFORMATION
(TASK NAME SUMMAY      fix hardware & components to door)
(ID                    )
(WBS                   )
(CONSTRAINTS           )
(TASK NAME             apply weatherstrip to door)
(DURATION              )
(START TIME            )
( FINISH TIME          )
(PREDECESSORS          )
(FIXED COST            £00)
(TOTAL COST            £100)
(RESOURCE NAMES         aluminum & vinyl weatherstrip,
                       carpenter, carpentery_equipment)
(COST/USE              £100)


RESOURCE SHEET
(RESOURCE NAMES        2*12 Douglas Fire Wood, carpenter,
                       carpentery_equipment)
(STANDARD RATE         £23)
(COST/USE              £100)


(RESOURCE NAMES     girt, clip, six #12*1 ¼" Self Drilling Screws,
                       carpenter, carpentry equipment)
(STANDARD RATE         £21)
(COST/USE              £100)

(RESOURCE NAMES        four#12*1½"Wood Lag Screws, carpenter,
                       carpentry equipment)
(STANDARD RATE         £21)
(COST/USE              £00)

(RESOURCE NAMES        metal panel,(22) #12*1 ¼" Self drilling
                        Screws, carpenter, carpentery_equipment)
(STANDARD RATE         £21)
(COST/USE              £100)


(RESOURCE NAMES        nine #12*1 ¼" Self drilling Screws,
                        carpenter, carpentery_equipment)
(STANDARD RATE         £21)
(COST/USE              £100)

(RESOURCE NAMES        doorjamb, (4)machine screws; (4)wood
                        lags, carpenter, carpentery_equipment)
(STANDARD RATE         £21)
```

```
(COST/USE             £100)

(RESOURCE NAMES    1¼"*8"h galvanized steel strut,(2)5/6" X
                ¾" nut & bolt, carpenter,carpentery_equipment)
(STANDARD RATE     £21)
(COST/USE          £100)

(RESOURCE NAMES       two 5/16" x ¾" hex head nut and bolt
                 screws  , carpenter, carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100)

(RESOURCE NAMES  three, 7 Gage, 10" x 1 ¼"" hinge
reinforcement, carpenter, carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100)

(RESOURCE NAMES       12 Gage galvanized steel angle,carpenter,
                       carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100)

(RESOURCE NAMES        4.86" x 1.75" x0.079" thick steel strike
                       plate,carpenter, carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100)

(RESOURCE NAMES       16 gage x 1 ¼"x6" strike plate
                      reinforcement?,carpenter,
                      carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100)

(RESOURCE NAMES       35 ¾"w x 831/8"h x 1 3/4"d 20G galvanized
                       steel doorleaf, carpenter,
                       carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100)

(RESOURCE NAMES     16 Gage flush top and bottom channel,
                       carpenter, carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100

(RESOURCE NAMES        (3) 9" x 1 ¼" x 7 Gage hinge
                       reinforcement, carpenter,
                       carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100)

(RESOURCE NAMES    16 Gage steel lock reinforcement plate,
                       carpenter, carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100)

(RESOURCE NAMES        (1) Yale 5307 Lever Lock, carpenter,
                       carpentery_equipment)
(STANDARD RATE        £21)
(COST/USE             £100)

(RESOURCE NAMES        4 ¼" steel Hinges, carpenter,
                       carpentery_equipment)
(STANDARD RATE        £21)
```

```
(COST/USE               £100)

(RESOURCE NAMES          aluminum & vinyl weatherstrip,
                        carpenter, carpentery_equipment)
(STANDARD RATE          £21)
(COST/USE               £100)
```

8.5.12 MS Project Scheduling data File in PSL Syntax & MS Project Terminology

The syntactic translator reads the Microsoft project Scheduling data file in Microsoft project representation as shown in **9.5.11** and produces a corresponding file using PSL syntax (KIF notation), but still persevering the Microsoft project Scheduling application's terminology. This file is the same as section as the file in **section 9.5.10** but includes Microsoft project specific scheduling data related to CCS cost estimating and AutoCAD design applications.

```
(<=>(=>(metal_bldg-opening  ?a)
      (summary task       ?a))
     (exists (?N)
       (=> (1000  ?N)
           (total cost   ?N ?a))))


(<=> (=>(construct wood frame  ?a)
       (task       ?a))
       (exists (?N)
              (=> (1000  ?N)
                 (total cost   ?N ?a))))


(<=> (=>(attach steel girt to wood frame   ?a)
       (summary task       ?a))
       (exists (?N)
              (=> (1000  ?N)
                 (total task cost   ?N ?a))))


  (<=> (=>(install metal panel  ?a)
       (summary task       ?a))
       (exists (?N)
              (=> (1000  ?N)
                 (total cost   ?N ?a))))


(<=> (=>(2*12 Douglas Fire Wood  ?r)
       (resource name      ?r))
       (exists (?N)
              (=> (250  ?N)
                 (cost/use  ?N ?r))))
(and (=>(girt, clip, six #12*1 ¼" Self Drilling Screws  ?r)
        (resource name             ?r))
     (exists (?N)
        (=> (250  ?N)
            (cost/use  ?N ?r))))


(<=>(=>(metal panel,(22) #12*1 ¼" Self drilling Screws  ?r)
        (resource name      ?r))
     (exists (?N)
       (=> (250  ?N)
           (cost/use  ?N ?r))))


(<=> (=>(construct wood frame  ?a)
        (task   ?a))
           (exists (?r1 ?r2 ?r3)
             (=>(and (=>(2*12 Douglas Fire Wood  ?r1)
                     (resource name     ?r1 ?a))
                 (=>(carpenter  ?r2)
                     (resource name     ?r2 ?a))
                   (=>(machine   ?r3)
                     (resource name     ?r3 ?a)))
           (exists (?N1 ?N2 ?N3)
                (and (=>(250  ?N1)
                       (cost/use  ?N1 ?r1))
                   (=>(200  ?r2)
                       (cost/use  ?N2 ?r2))
                     (=>(300    ?r3)
                       (cost/use  ?N3 ?r3)))))))


(<=> (=>(attach girt to wood frame   ?a)
        (summary task        ?a))
    (exists (?r1 ?r2 ?r3 ?r4)
       (=>(and (=>(girt, clip, six #12*1 ¼" Self Drilling Screws
?r1)
                 (resource name   ?r1 ?a))
               (=>(four #12 *1 ½"Wood Lag Screws  ?r2)
                  (resource name   ?r2 ?a))
                (=>(carpenter  ?r3)
                     (resource name    ?r3  ?a))
                 (=>(machine   ?r4)
                    (resource name   ?r4  ?a)))
       (exists (?N1 ?N2 ?N3)
                (and (=>(250  ?N1)
                       (cost/use    ?N1   ?r1))
                    (=>(200  ?r2)
                       (cost/use    ?N2 ?r2))
                     (=>(300    ?r3)
                       (cost/use  ?N3 ?r3))
                      (=>(400    ?r4)
                       (cost/use    ?N4 ?r3)))))))


(<=> (=>(attach metal panel to vertical member & doorframe  ?a)
        (summary task      ?a))
  (exists (?r1 ?r2 ?r3 ?r4)
        (=>(and (=>(metal panel,(22) #12*1 ¼" Self drilling
Screws ?r1)
               (resource name  ?r1  ?a1 ))
            (=>(nine #12*1 ¼" Self drilling Screws?r2)
               (resource name  ?r2  ?a2))
             (=>(carpenter  ?r2)
                (resource name    ?r2   ?a1 ?a2))
             (=>(machine   ?r3)
                (resource name     ?r3    ?a1 ?a2)))
  (exists (?N1 ?N2 ?N3)
               (and (=>(250  ?N1)
                      (cost/use    ?N1   ?r1))
                   (=>(200  ?r2)
                      (cost/use    ?N2 ?r2))
                    (=>(300    ?r3)
                      (cost/use  ?N3 ?r3))
                     (=>(400    ?r4)
                     (cost/use    ?N4 ?r3)))))))
```

8.5.13 MS Project Scheduling Data File in PSL Representations

The semantic translator (semantic translation definitions) for MS project Scheduling application reads the MS project Scheduling data file written in the application's terminology and PSL (KIF notation) syntax (**section 9.5.12**) and produces a corresponding file containing only PSL terminology in PSL syntax (KIF) by substituting the definition of all the MS project Scheduling application's terminologies with definitions in PSL. Using the translation definitions for MS Project Scheduling the scheduling data file in PSL syntax (KIF) and the application's terminology of **section 9.5.12** is mapped to PSL definitions that contain only PSL terminology.

```
(<=>(=>(metal_bldg-opening  ?a)
      (activity      ?a))
     (exists (?N)
       (=> (1000  ?N)
           (cost of activity   ?N ?a))))

(<=> (=>(construct wood frame  ?a)
       (activity      ?a))
       (exists (?N)
              (=> (1000  ?N)
                 (cost of activity   ?N ?a))))


(<=> (=>(attach steel girt to wood frame   ?a)
       (activity      ?a))
       (exists (?N)
              (=> (1000  ?N)
                 (cost of activity   ?N ?a))))


  (<=> (=>(install metal panel  ?a)
       (activity      ?a))
       (exists (?N)
              (=> (1000  ?N)
                 (cost of activity   ?N ?a))))
```

```
(<=> (=>(2*12 Douglas Fire Wood  ?r)
       (consumable      ?r))
      (exists (?N)
              (=> (250  ?N)
                  (cost of resource  ?N ?r))))


(and (=>(girt, clip, six #12*1 ¼" Self Drilling Screws  ?r)
        (consumable      ?r))
     (exists (?N)
        (=> (250  ?N)
            (cost of resource  ?N ?r))))


(<=>(=>(metal panel,(22) #12*1 ¼" Self drilling Screws  ?r)
       (consumable   ?r))

(exists (?N)
       (=> (250  ?N)
           (cost of resource  ?N ?r))))


(<=> (=>(construct wood frame  ?a)
       (activity   ?a))
     (exists (?r1 ?r2 ?r3)
             (=>(and (=>(2*12 Douglas Fire Wood  ?r1)
                        (consumable     ?r1 ?a))
                     (=>(carpenter  ?r2)
                        (reusable    ?r2 ?a))
                     (=>(machine   ?r3)
                        (reusable    ?r3 ?a)))
           (exists (?N1 ?N2 ?N3)
              (and (=>(250  ?N1)
                         (cost of resource  ?N1 ?r1))
                       (=>(200  ?r2)
                         (cost of resource  ?N2 ?r2))
                       (=>(300   ?r3)
                         (cost of resource  ?N3 ?r3)))))))


(<=> (=>(attach girt to wood frame   ?a)
       (activity        ?a))
    (exists (?r1 ?r2 ?r3 ?r4)
       (=>(and (=>(girt, clip, six #12*1 ¼" Self Drilling Screws
?r1)
                  (consumable   ?r1 ?a))
               (=>(four #12 *1 ½"Wood Lag Screws  ?r2)
                  (consumable   ?r2 ?a))
                (=>(carpenter  ?r3)
                     (reusable    ?r3  ?a))
                  (=>(machine   ?r4)
                    (reusable   ?r4  ?a)))
        (exists (?N1 ?N2 ?N3)
                (and (=>(250  ?N1)
                         (cost of resource    ?N1   ?r1))
                    (=>(200  ?r2)
                         (cost of resource    ?N2 ?r2))
                      (=>(300    ?r3)
                         (cost of resource  ?N3 ?r3))
                       (=>(400    ?r4)
                         (cost of resource    ?N4 ?r3)))))))
```

```
(<=> (=>(attach metal panel to vertical member & doorframe  ?a)
       (activity      ?a))
  (exists (?r1 ?r2 ?r3 ?r4)
       (=>(and (=>(metal panel,(22) #12*1 ¼" Self drilling
Screws ?r1)
              (consumable  ?r1  ?a1 ))
           (=>(nine #12*1 ¼" Self drilling Screws?r2)
              (consumable  ?r2  ?a2))
            (=>(carpenter  ?r2)
               (reusable    ?r2    ?a1 ?a2))
             (=>(machine   ?r3)
               (reusable    ?r3    ?a1 ?a2)))
  (exists (?N1 ?N2 ?N3)
               (and (=>(250  ?N1)
                      (cost of resource    ?N1   ?r1))
                   (=>(200  ?r2)
                      (cost of resource    ?N2 ?r2))
                    (=>(300    ?r3)
                      (cost of resource  ?N3 ?r3))
                     (=>(400    ?r4)
                        (cost of resource    ?N4 ?r3)))))))
```

Chapter 9 Discussion

9.1 Introduction

This research study has ınvestıgated the applıcabılity of PSL, a standard Process Specıficatıon and exchange Language for manufactunng applıcations interoperabılıty, in the construction environment and ımplemented the language ın a construction scenano.

Thıs chapter presents the discussıon on the investıgatıon of the applıcabılity of the Process Specıfication Language ın the constructıon environment and the implementatıon of the language in a construction scenano withın the pre-constructıon process stages and the outcome of the research work ıs summarised.

9.2 Investigation of the Applicability of PSL in Construction

Pnor to the use of the PSL language ın the constructıon scenario, for the case study project constructıon process informatıon exchange between the scenano software applıcatıons, the applıcabılıty of the language ın constructıon was ınvestıgated In thıs ınvestıgation the applıcatıons' process ınformation representatıons' semantıc concepts were analysed and mapped to the manufactunng process concepts defined ın the PSL ontology to determıne how applıcable this language is ın the construction environment.

PSL ontology ıs a map of defınıtions for the semantıc concepts of manufacturing processes developed through mapping of the process concepts related to vanous manufactunng applıcations along wıth theır semantıc defınıtions. The semantıc concepts of manufactunng processes are defined explıcıtly and clearly using Knowledge Interchange Format (KIF) (Genesereth, M. and Fikes, R. 1992). In **chapter 6, figure 6.4** shows the PSL ontology parts that are expected to contain concepts for defining the constructıon process informatıon representations (desıgn, cost estımatıng and scheduling data types) wıthın the pre-constructıon process stages. The constructıon process ınformatıon representations wıthın the pre-constructıon process stages were identıfied through a technical analysis of the AutoCAD, CCS, and Mıcrosoft project schedulıng applıcations. The concept maps (conceptual models) for the design, cost estimatıng, and scheduling applications shown in **figures 6.5 to 6.7** were developed through mapping of the AutoCAD, CCS, and Microsoft project applications' data types. In order to conduct multidimensional conceptual analysis between the applications' information representation data types and PSL ontology, and based on the result of this analysis to map the applications' data types of these maps, to the PSL ontology parts shown in **figure 6.4**, the semantic concepts of these data types needed to be analyzed and defined. The concept maps of **figures 6.5 to 6.7** provided a framework within which theories can be stated, that postulates relationships between concepts or data types. Based on this, the semantic concepts of the three applications' data types were defined using a natural language statements and was used for conducting multidimensional conceptual analysis of the construction process concepts in correlation with the PSL ontology concepts. These definitions also served as a reasoning basis during the mapping of the applications data types to the PSL ontology to select the PSL parts that contain applicable semantic concepts for defining the applications' data types.

In the concept mapping of the construction process information representations within the pre-construction process stages (that are the AutoCAD, CCS, and Microsoft project applications' data types) to PSL ontology, the natural language definition of the semantic concepts of the applications' data types developed in **section 6.4.3** provided the reasoning basis for identifying and selecting the PSL parts and concepts that are applicable in the pre-construction process stages and the need for new extensions to incorporate the concepts that are identified to be new to the PSL ontology.

The concept mapping method, allowed the development of semantic theories from the applications concept maps that are used in the identification of PSL parts applicable in the pre-construction process stages, and the need for new extensions and provided the capability to connect new ideas to that already known concepts of PSL parts and to organize them in a logical structure.

In the investigation of the applicability of PSL in construction, the data types within each of the applications maps shown in **figures 6.5 to 6.7** were mapped to the PSL ontology shown in **figure 6.4** based on the result of multidimensional conceptual analysis between the natural language definitions of the semantic concepts of the applications data types and the manufacturing process concepts' definitions within the PSL ontology. Based on the result of multidimensional conceptual analysis between the natural language definitions of the semantic concepts of AutoCAD, CCS, and Microsoft project applications' data types and the manufacturing process concepts' definitions within the PSL ontology, the "Processor Activities" of part 46, "*Sub-Activity Theory*" of part 12, "Theory of *Resource Requirements*" of part 14, and "*Reasoning about Resource Divisibility*" of part 44 of PSL ontology were identified to contain the applicable definitions for the three applications data types semantic concepts. During the mapping of the applications data types to the PSL ontology shown in **figure 6.8 to 6.10,** the need for new extensions to the PSL ontology to incorporate the semantic concept of "construction products, or construction project design elements" for AutoCAD design application "resources categories, billed quantities and cost" for CCS application and Microsoft project application were identified. The concepts and definition (theories) of the PSL parts 12,14,44 and 46, of the PSL ontology are as follows:

Processor Activities

(http://www.mel.nist.gov/psl/psl-ontology/part46/processor.html)

Extension Name: processor.def
Primitive Lexicon: None
Defined Lexicon:

- (modifies ?a ?r)
- (processor_activity ?a)
- (processor_resource ?r ?a)
- (input_material ?r ?a)
- (output_material ?r ?a)

Theories Required by this Extension: requires.th, act_occ.th, complex.th, atomic.th, occtree.th, disc_state.th, subactivity.th, psl_core.th

Definitional Extensions Required by this Extension: res_role.def

Overview
In this extension, we are not dealing with arbitrary activities; the manufacturing activities use and consume resources to produce or modify resources, ultimately creating some product. This class is axiomatized using the term ``processor activity''.

Definitions

Definition 1 An activity ?a modifies an object ?r if there is some property of the resource which is changed as the result of the occurrence of ?a

```
(defrelation modifies (?a r) :=
(exists (?f)
        (and    (resource_fluent ?f ?r)
                (changes ?a ?f))))
```

Definition 2 *A processor activity is an activity which uses some set of resources, consumes or modifies some other set of resources, and produces or modifies a set of objects.*

```
(defrelation processor_activity (?a) :=
(exists (?r1 ?r2 ?r3)
        (and    (or     (reusable ?r1 ?a)
                        (possibly_reusable ?r1 ?a))
                (or      (consumable ?r2 ?a)
                        (possibly_consumable ?r2 ?a)
                         (modifies ?a ?r2))
                (or      (creates ?a ?r3)
                         (modifies ?a ?r3)))))
```

Definition 3 *An object ?r is a processor resource for an activity ?a iff ?a is a processor activity which uses ?r.*

```
(defrelation processor_resource (?r ?a) :=
(and    (processor_activity ?a)
        (reusable ?r1 ?a)
        (possibly_reusable ?r1 ?a)))
```

Definition 4 *An object ?r is an input material for an activity ?a iff ?a is a processor activity which consumes or modifies ?r.*

```
(defrelation input_material (?r ?a) :=
(and    (processor_activity ?a)
        (or     (consumable ?r ?a)
               (possibly_consumable ?r ?a)
                (modifies ?a ?r))))
```

Definition 5 *An object ?r is an output material for an activity ?a iff ?a is a processor activity which produces or modifies ?r.*

```
(defrelation output_material (?r ?a) :=
(and    (processor_activity ?a)
        (or     (creates ?a ?r)
                (modifies ?a ?r))))
```

Theory of Subactivities

(http://www.mel nist.gov/psl/psl-ontology/part12/processor.html)

The PSL Ontology uses the subactivity relation to capture the basic intuitions for the composition of activities.

The core theory subactivity.th alone does not specify any relationship between the occurrence of an activity and occurrences of its subactivities. For example, the specification of subactivities alone does not allow us to distinguish between a nondeterministic activity and a deterministic activity.

The basic ontological commitments of the Subactivity Theory are based on the following intuitions:

Intuition 1:
The composition relation is a discrete partial ordering, in which primitive activities are the minimal elements.

Informal Semantics for the Subactivity Theory
(subactivity ?a1 ?a2) is TRUE in an interpretation of Subactivity Theory if and only if activity ?a1 is a subactivity of activity ?a2.

(primitive ?a) is TRUE in an interpretation of Subactivity Theory if and only if the activity ?a has no proper subactivities.

Theory of Resource Requirements

(http //www mel nist.gov/psl/psl-ontology/part14/processor html)

Extension Name: requires.th
Primitive Lexicon:

- (requires ?a ?r)

Theories Required by this Extension: act_occ.th, complex.th, atomic.th, subactivity.th, disc_state.th, occtree.th, psl_core.th

Defined Lexicon:

- (resource ?r)
- (interfering ?a1 ?a2)
- (common ?r ?a1 ?a2)

Definitional Extensions Required by this Extension: None

Axioms

Axiom 1: If two activities are interfering, there exists a resource that they share.

```
(forall ?a1 ?a2)
        (=>     (interfering ?a1 ?a2)
                (exists (?r)
                        (and    (requires ?a1 ?r)
                                (requires ?a2 ?r))))))
```

Supporting Definitions

A resource is any object which is required by some activity.

```
(defrelation resource (?r) :=
(and    (object ?r)
        (exists (?a)
                (requires ?a ?r))))
```

This relation is satisfied iff ?r is the only resource shared by activities ?a1 and ?a2, that is, it is the only resource which is required by both activities.

```
(defrelation common (?r ?a1 ?a2) :=
(forall (?r2)
    (=>    (and    (requires ?a1 ?r2)
                   (requires ?a2 ?r2))
         (= ?r2 ?r))))
```

Reasoning about Resource Divisibility

(http://www.mel.nist.gov/psl/psl-ontology/part44/processor.html)

Extension Name: res_divisible.def
Primitive Lexicon: None

Defined Lexicon:

- (consumes_quantity ?a ?r ?q)
- (produces_quantity ?a ?r ?q)
- (uses_quantity ?a ?r ?q)
- (creates ?a ?r)
- (destroys ?a ?r)
- (uses ?a ?r)
- (consumes ?a ?r)
- (produces ?a ?r)
- (provides ?a ?r)
- (provides_quantity ?a ?r ?q)

Theories Required by this Extension: requires.th, act_occ.th, complex.th, subactivity.th, atomic.th, disc_state.th, occtree.th, psl_core.th

Definitional Extensions required by this Extension: none

Definitions

It is important to realize that the notion of quantity referred to in this extension is independent of the notion of resource existence or identity, as well as the distinction between discrete and continuous resources. We are only concerned with the availability of the resource for a future activity. The notion of resource point is a constraint on such availability. The quantity specifies the divisibility of the resource with respect to multiple concurrent activities. The resource point determines the maximal set of concurrent activities which require the resource. Thus, if the quantity is zero, then no activity that requires the resource is possible. **Definition 1** An activity consumes some quantity ?q of a resource iff the demand for the resource is ?q and the resource point of the resource is decremented by ?q after the occurrence of the activity.

```
(defrelation consumes_quantity (?a ?r ?q) :=
(forall (?q1 ?occ ?occ1 ?occ2)
        (=>     (and    (do ?a ?occ1 ?occ2)
                        (holds (demand ?a ?r ?q) ?occ1)
                        (holds (resource_point ?r ?q1) ?occ1))
                (holds (resource_point ?r (- ?q1 ?q)) ?occ2)))))
```

Definition 6 An activity creates a resource if it produces some quantity of the resource, and the quantity of the resource before the occurrence of the activity was zero.

```
(defrelation creates (?a ?r) :=
(exists (?q1)
        (and    (produces_quantity ?a ?r ?q1)
                (forall (?q2 ?occ)
                        (=>     (and    (occurrence_of ?occ ?a)
                                        (prior   (resource_point   ?r   ?q2)
?occ)
                                (= ?q2 0))))))
```

Definition 7 An activity destroys a resource if it consumes some quantity of the resource, and the quantity of the resource after the occurrence of the activity is zero.

```
(defrelation destroys (?a ?r) :=
(exists (?q1)
        (and    (consumes_quantity ?a ?r ?q1)
                (forall (?q2 ?occ)
                        (=>     (and    (occurrence ?occ ?a)
                                        (prior  (resource_point  ?r  ?q2)
?occp)
                                (= ?q2 0))))))
```

Definition 9 *An activity uses a resource if it uses some quantity of the resource.*

```
(defrelation uses (?a ?r) :=
(exists (?q)
        (uses_quantity ?a ?r ?q)))
```

Definition 10 *An activity consumes a resource if it consumes some quantity of the resource.*

```
(defrelation consumes (?a ?r) :=
(exists (?q)
        (consumes_quantity ?a ?r ?q)))
```

Definition 12 *An activity produces a resource if it produces some quantity of the resource.*

```
(defrelation produces (?a ?r) :=
(exists (?q)
        (produces_quantity ?a ?r ?q)))
```

Definition 14 *An activity provides some quantity of a resource if there exists a subactivity that produces some quantity of the resource and another subactivity that consumes some quantity of the resource.*

```
(defrelation provides_quantity (?a ?r ?q) :=
(and    (exists (?a1)
                (and    (subactivity ?a1 ?a)
                        (produces_quantity ?a1 ?r ?q)))
        (exists (?a2)
                (and    (subactivity ?a2 ?a)
                        (consumes_quantity ?a2 ?r ?q)))))
```

Definition 15 *An activity provides a resource if it provides some quantity of a resource.*

```
(defrelation provides (?a ?r) :=
(exists (?q)
        (provides_quantity ?a ?r ?q)))
```

9.3 Implementation of PSL in the Pre-Construction Process Stages

The trial implementation of PSL in construction proceeded in the same way; the PSL language was implemented within the manufacturing environment, but in a different environment by exploring a construction scenario within the pre-construction process stages. After the initial development of PSL, from a single scenario there were multiple pilot implementations of the language each focusing on different applications within the manufacturing environment, to further expand the language to incorporate the process concepts introduced in various manufacturing applications when PSL was used to exchange process information between the applications. The purpose of PSL implementation in the construction scenario was not to propose changes to or extend the PSL ontology but: (1) to evaluate the degree of applicability of the concepts and definitions of the language in providing common definitions to the construction process concepts shared between the scenario applications, (2) to evaluate the use and the impact of this standard language in construction practices within the pre-construction process stages, (3) to asses the impact of the construction process concepts within the pre-construction process stages in the use of the language in construction, and (4) to identify the construction process concepts that cannot be completely defined using the current PSL parts and concepts or those concepts which are new to the PSL ontology and propose concepts and theory related to these concepts for new extensions development to the PSL ontology and specify the relations that should exist between the existing PSL process concepts and the new proposed concepts.

9.3.1 The Construction Scenario

The construction scenario involves process related design, pre-tender cost estimating, and scheduling or planning information representation for construction of a case study project and exchange of the process information between construction computer applications supporting the pre-construction process stages. A case study of construction project "a metal door in a metal opening of a building" (the textual construction specification for the case study project shown in **tables 8.1 and 8.2** and the drawings of the project products attached in **Appendix E** were obtained from Hurricane Engineering and Testing INC.) was identified for this scenario, and AutoCAD, CCS, and Microsoft project software were selected as candidate scenario software applications to represent the case study project construction process information and exchange this information using PSL.

The reasons behind choosing the construction scenario within the pre-construction stages; the AutoCAD, CCS, and Microsoft project software applications to support the scenario and the "a metal door in a metal opening of a building project" as a case study project for the construction scenario for the trial implementation of PSL in the construction environment are as follows:

- **The construction scenario:** the construction scenario within the pre-construction process stages was chosen as a test ground for the trial implementation of PSL in construction because (1) it is in these stages that most of the construction processes are described, (2) these stages have exhibited complex process information representation and exchange and, (3) most of the computer applications supporting theses process stages use different terms and representation to specify the same information.
- **AutoCAD, CCS, and Microsoft project:** these packages were selected as test applications for the construction scenario The primary areas in which computer tools are widely used to support construction processes descriptions and management are project design, cost estimating and planning and scheduling. To data, there are various computer programs that can be used within these areas. Amongst the other AutoCAD, Microsoft Project and CCS packages were selected as test applications for the trial implementation of PSL in construction. The reasons behind the choice of these applications are: AutoCAD and Microsoft Project application are the most widely used applications in the UK construction organisations, and CCS is an application with complex information representation structure. As shown in **sections 6.3.1.1-6.3.1.3** different computer applications have different information representation structures with main entities and associated attributes organised differently. Hence how information is structured in computer applications is important criteria in determining how complex the applications information representations are and as well as the information represented As shown from the CCS conceptual model in **figures 6.6 (a & b)** CCS has complex data structure, additionally it uses different file formats for input and output data files. The combination of these packages provided sound test applications for the trial implementation of PSL in construction and complex construction process information representation structures for evaluating the impact of the language in construction process information exchange and the impact of the construction process concepts, defined by the organisation of the applications' information representations structures, on the use of PSL in construction.
- **The construction case study project:** the implementation of PSL in the construction scenario required a construction case study project, and a "metal door in a metal opening of a building project" was selected in order to have a real construction process information exchange, between the scenario software

applications, using PSL as interchange language. These types of doors are mostly used for buildings in hurricane vulnerable areas and their construction detail is complex. This type of project with complex construction and information has been selected for the construction scenario to illustrate the complex process related information in construction projects that need to be represented unambiguously, shared and exchanged between applications consistently, and this kind of project provides a strong test case and complex process information for the trial implementation of PSL in the construction environment and evaluation of the impact of the language in process information exchange between construction applications, and o the construction process concepts on the use of the langue in construction.

9.3.2 Capability of the Scenario Applications to Exchange Information

Form the technical analysis of the scenario software applications, it was identified, as described in **section 7.4.2**, that in addition to the native file formats theses applications read from and write to other file formats and as summarised in **table 7.8,** some of these are common to the three applications. For example CCS and Microsoft project scheduling applications read from and write to Excel files, and AutoCAD allows a designer to extract attribute information from drawings of projects in CDF (comma delimited format); SDF (space delimited format), or DXF (data exchange format) file formats, that can be read by many popular database management programs. The CDF, SDF and DXF file formats can be imported into any word processing program that accepts ASCII files, these include, Microsoft Excel and Microsoft Access files. Hence the three scenario software applications can exchange their respective files (import and export their respective physical files) using Excel as a common exchange file format. However, the exchange of physical files facilitated by the common file format, Excel is not sufficient for interoperability of the scenario applications. This is because, as learned from the data types and their semantic concepts identified during the technical analysis of the scenario software applications and conceptual analysis of these data types', these applications use different terms to represent the same information or the same terms may not convey the same information and this can lead to incorrect information exchange between the software applications. As a result it was identified that, for correct information exchange between these applications require a common standard language that can resolve the semantic differences that exist between some of the applications' information representations (terminologies). In this research study PSL is identified as a promising language for bridging the semantic gaps and consistent process information exchange between construction software applications. In this way the applications can import and export the correct and meaningful information through one of the common exchange file formats such as the Excel file using PSL as a common language between the applications information representations, terminology in these files. Hence PSL based syntactic and semantic translation programs need to be developed to translate information from and to the common exchange file format (Excel) and PSL, the applications can then import and export the information in this common exchange file.

9.3.3 IDEF3 Modelling of the Case Study Project Construction Processes

In the description of the construction case study project "the metal door in a metal opening of a building project" an IDEF3 model of the project construction processes was developed, based on the textual construction specifications (shown in **tables 8.1 & 8.2**) of the case study project. This was to represent the construction processes and process related information in a formal way and to identity data related to design, cost estimating, and scheduling applications for each activity (activities are processes at the lowest level of decomposition of the higher level process in the process model) of the process model The IDEF3 construction process model (shown in **section 8.3)** represented the construction processes, material of construction, and construction products information of the case study project. This is process information related to design, which is extracted from the case study project construction specification shown in **tables 8.1 & 8.2** in correlation with the drawings attached in **appendix E.** Data related to cost estimating and scheduling could also be gathered for each activity of the process model. However to keep the model clear the data related cost estimating and scheduling are represented using CCS and Microsoft project applications process information representations based on the design related process information represented in the IDEF3 model.

In the representation of the construction process information for the case study project the IDEF3 Process Description Capture method was chosen because, among the process modeling methodologies available to date, IDEF3 Process Capture Method best meets most of the process characteristics representation requirements. IDEF3 provides graphical constructs to represent processes and sub processes, relationships between processes at the same and different level of abstraction, sequences and dependencies between processes, splitting of processes into other sub-processes, synchronisation and asynchronisation of processes, resources requirements of each activity of the model, and it is even possible to attach data such as time and other constraint to each constructs of the model. From the model it is proven that the method is a well-structured technique for describing processes from scheduling and planning point of view. IDEF3, like others is a diagramming language but provides more graphical constructs and a set of forms (elaboration forms) for capturing any information about the processes and participant objects involved In this way construction process information for the case study project, related to the three scenario candidate applications have been captured

However, IDEF3 is one of the process representations that focus on the syntax of process specifications rather than the meaning of terms, the semantics. These models lack an adequate specification of the semantics of the process terminology, which leads to inconsistent interpretations and uses of information. While such representation is sufficient for exchanging information between applications of the same type, e.g. scheduling applications, different types of applications associate different meanings with similar or identical terms Hence the IDEF3 representation of the construction process information, related to design, cost estimating and scheduling applications, for the case study project was not sufficient to support the information exchange between AutoCAD design, CCS cost estimating and Microsoft project scheduling scenario software applications. This is because the three applications use different terms to mean the same or same terms may not convey the same information. Without explicit definitions of the terms, it is difficult to see how concepts in each of these applications correspond to each other Simply sharing terminology is insufficient to support interoperability; the applications must share their semantics, i e., the meanings of their respective terminology. Such representations require a formal specification of the semantics of terminologies. One approach to generating this specification is through the use of ontology. Ontology provides unambiguous definitions for terminology that helps to capture key distinctions among concepts in the different applications, which aids in the translation process However the conditions that may put on acceptable models/ontology are derived from descriptions of a domain or a system that is the data from which the model is constructed. Hence IDEF3 descriptions are basis for modelling such as in the ontology building process. To overcome this problem, as shown in **chapter 8,** PSL a formal ontology based standard language was implemented in the construction scenario for the "metal door in metal opening of a building project construction process information exchange between AutoCAD, CCS, and Microsoft project applications.

9.3.4 Software Applications' Representation of the Process Information

Based on the IDEF3 process descriptions (process information related to design) of the case study project, a complete construction process related design, cost estimating and scheduling information were developed using AutoCAD design, CCS cost estimating, and Microsoft project scheduling scenario software applications' textual information representation formats. The screen shots, of the three-scenario software applications' representation of the case study project construction process information are shown in **figures 8.13, 8.14(a & b), and 8.15 (a & b).** These representations provided the complete construction process information for the case study project and basis for the development of the native ontology (the applications process concepts definitions) of each of the scenario software applications. The data types of each of these applications and the corresponding process related data of the case study project represented by these data types determined the native ontology of each of the scenario applications.

9.3.5 Information Exchange between the Scenario Applications using PSL

From the technical analysis of the data types of the software applications, AutoCAD, CCS and Microsoft project scheduling, it was learned that these applications use different terminology to represent the same information or the same terms may not represent the same information. Therefore exchanging information using one of the common exchange file formats identified in **chapter 7** would lead to exchange of incorrect information. As one of the main objectives of this research study PSL, the emerging standard language developed for the manufacturing sector, was implemented in the construction scenario to allow the three candidate software applications, exchange the exact semantics of their information representations (terminology) of the case study project construction processes In this implementation PSL was used as a common interchange language to resolve the semantic differences between the scenario software applications' information representations or terminologies.

For consistent and correct translation of information representations between scenario software applications using PSL, information exchange required to be between the ontologies of the applications and PSL. The data types of each of the scenario software applications and the corresponding process related data of the case study project determined the native ontology of each of the scenario applications as shown in **sections 8.5.2, 8.5.6 & 8.5.7 and 8.5.11**. The native ontology of each of the applications provides the basis from and to which information must to be translated for information exchange between the applications using PSL. During information exchange between the applications using PSL as an interchange language, the syntactic and semantic translations of process information representations between the applications and PSL were carried out from the applications ontology to the PSL ontology and back to the applications ontology as shown in **sections 8.5.2 to 8.5.13.** The syntactic translation is a parsing between the PSL syntax (KIF notation) and the native syntax of the applications, this parsing keeps the original terminology of each of the applications and produce KIF representations of the applications information representations (the applications native ontologies). For the semantic translations between the scenario software applications information representations (terminology) hand written translators (semantic translation definitions) were developed between each of the scenario software applications and PSL as shown in **section 8.5.1.**

The semantic translators (semantic translation definitions) shown in section 8.5.1 for AutoCAD, CCS, and Microsoft project scheduling applications were developed based on the applications and PSL terminologies to allow the translation of information between the applications using PSL These translation definitions between each of the scenario software applications ontology and PSL were driven by the ontological definitions that were written using the same foundational theories. These are definitions for the terminology of the application ontology, using only the terminology from the PSL Ontology, as well as definitions for the terminology of the PSL Ontology using only the terminology of the application ontology.

In the implementation of PSL in the construction scenario, the concepts from PSL parts that were selected (during the investigation of the applicability of PSL in construction) to be applicable to resolve the semantic differences between the scenario software applications were used in the development of the semantic translation definitions for process information translation between the applications representations. The semantic information translation between the scenario applications were based on the KIF (or PSL syntax) representations of the applications native ontology. This is because KIF representation of each of the applications native ontology provides a formal representation of the applications process information representations that can be mapped to the PSL process concepts representations syntax and semantics. The KIF representations of each of the applications native ontology as shown in **sections 8.5.3, 8.5.7 & 8.5.8 and 8.5.12,** were determined by the natural language theories developed for the scenario applications data types' semantic concepts in **sections 6.4.3** for each of the applications concepts maps. **Tables 9.1 (a, b, & c)** summarised the PSL parts and theories, and KIF representations of each of the applications' native ontology.

PSL Parts	**PSL Theories**	**Applications ontology in KIF syntax** **AutoCAD Design**
	There is no concepts defined in PSL	(forall (?x) (=> (dwg_object ?x) (exists (?x1 .) (components ?x. ?x))))
Processor Activity Part 46	(defrelation input_material (?r ?a) = (and (processor_activity ?a) (or (consumable ?r ?a) (possibly_consumable ?r ?a) (modifies ?a ?r))))	(forall (?x1) (=> (dwg_object_component ?x1 ?x) (exists (?a) (construction ?a ?x1) (exists (?r1) (material ?r1 ?a)))))

Table 9.1 (a). PSL parts and theories used in the information translations process to capture the AutoCAD design related process information

PSL Parts	**PSL Theories**	**Applications ontology in KIF syntax** **CCS Cost Estimating**
Theory of Subactivities part 12	**Macro_ops which is equivalent to dwg_object in AutoCAD are not defined in PSL. But macor_op and simple_op as activities are defined in the sub_acivity theory**	(forall (?a) (=> (macor_op ?a) (exists (?a1) (macor_op ?a1 ?a) (exists (?a2...) (sub_macor_op ?a2... ?a1)
Processor Activity Part 46	(defrelation processor_activity (?a) = (exists (?r1 ?r2 ?r3) (and (or (reusable ?r1 ?a) (possibly_reusable ?r1 ?a)) (or (consumable ?r2 ?a) (possibly_consumable ?r2 ?a) (modifies ?a ?r2)) (or (creates ?a ?r3) (modifies ?a ?r3)))))	(forall (?a) (=> (simple_op ?a) (exists (?ar1 ?r2 ?r3 ?r4 ?r5) (=> (and (plant ?r1 ?a) (plant ?r1 ?a) (material ?r2 ?a) (labour ?r3 ?a) (temp_material ?r4 ?a) (subcontract ?r5 ?a)) (require ?a ?r)))))
	Cost rate and utilization of resources are not defined in PSL	(forall (?r) (=> (or (plant ?r) (material ?r) (labour ?r) (temp_material ?r) (subcontract ?r)) (exists (?c ?z) (and (rate ?c ?r) (utilization ?z ?r)))))
	Nett_raes and Nett_amount of macro and simple operations are not defined in PSL	(forall (?a) (=> (or (macro_op ?a) (simple_op ?a)) (exists (?Nr Nm) (and (Nett_rate ?Nr ?a) (Nett_amount ?Nm)))

Table 9.1(b) PSL parts and theories used in the information translations process to capture the CCS cost estimating related process information

PSL Parts	PSL Theories	Applications ontology in KIF syntax Microsoft Project Scheduling
Theory of Subactivities part 12	**(subactivity ?a1 ?a2)** is TRUE in an interpretation of Subactivity Theory if and only if activity ?a1 is a subactivity of activity ?a2.	(forall (?a) (=> (summary_task ?a) (exists (?a1...) (sub_tasks ?a1... ?a))))
	Fixes and total cost of tasks and summary tasks and are not defined in PSL	(forall (?a) (=> (or (summary_task ?a) (task ?a)) (exists (?Ct ?Cf) (and (total_cost ?Ct ?a) (fixed_cost ?Cf ?a)))))
Processor Activity Part 46	(defrelation processor_activity (?a) := (exists (?r1 ?r2 ?r3) (and (or (reusable ?r1 ?a) (possibly_reusable ?r1 ?a)) (or (consumable ?r2 ?a) (possibly_consumable ?r2 ?a) (modifies ?a ?r2)) (or (creates ?a ?r3) (modifies ?a ?r3)))))	(forall (?a) (=> (task ?a) (exists (?r1..) (=> (resources ?r1...) (require ?a ?r1)))))
	Cost/use and standard rates of resources are not defined in PSL	(forall (?r) (=> (resource ?r) (exists (?Cu ?Cs) (and (cost/use ?Cu ?r) (standard_rate ?Cs ?r)))))
	Cost/use and standard rates of resources and total and fixed cost of tasks are not defined in PSL	(forall (?a) (=> (task ?a) (exists (?r1 .) (=>(=> (resources ?r1) (require ?a ?r1.)) (exists (?Cu ?Cs) (=> (and (cost/use ?Cu ?r) (standard_rate ?Cs ?r)) (exists (?Ct ?d) (=> (and (total_cost ?Ct ?a) (duration ?d ?a)) (or (total_cost (sum_of ?Cu) ?a) (total_cost (* ?Cs ?d) ?a))))))))))

Table 9.1. (c) PSL parts and theories used in the information translations process to capture the Microsoft project scheduling related process information

9.3.6 Data Translation between Scenario Applications Representations

Based on the semantic translation definitions (hand written translators) developed between AutoCAD, CCS, and Microsoft project scheduling applications and PSL (as shown in **section 8.5.1**) and using parsing method, the result of hand written syntactic and semantic translations that were carried out to exchange, the case study project construction process information, between the three scenario software applications is shown in **sections 8.5.2 – 8.5.13**. In this translation a two-stage process, syntactic (syntactic parsing) and semantic (semantic mapping) translations between the applications and PSL representations were conducted. The syntactic parser reads data from the applications exchange file (or the application's native ontology) and generates a corresponding file using PSL syntax, which is a Knowledge Interchange Format (KIF) notation, but still preserving the applications terminology, and a reverse syntactic parsing reads KIF representation of the applications data and writes back to the applications exchange file (or the application's native ontology). The semantic translator reads KIF representation of one of the applications data and translates to PSL syntax and terminology and/or reads a file in PSL syntax and terminology and translates to KIF representation of one of the applications data **Figure 9.1** illustrates the flow of semantic information, from the applications representations to PSL and back to the applications, during the process information representations translations between the scenario applications using PSL.

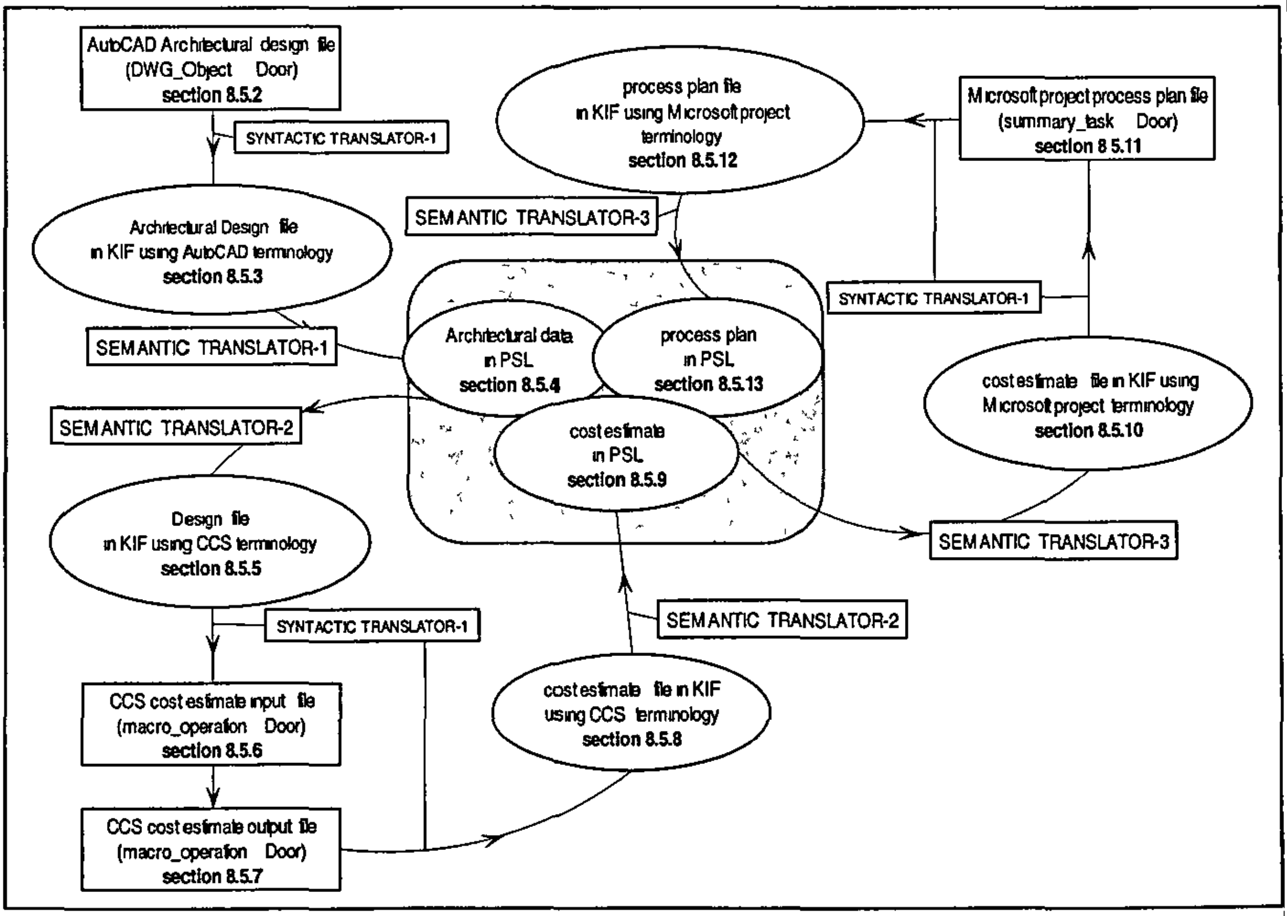


Figure 9.1. Flow of semantic information between scenario applications and the role of PSL

The section numbers of the files are shown in **figure 9.1** to illustrate the flow of information between the applications and the role of PSL when reading the translation process in the following section. These sections represent the process information and their representations as follows:

8.5.2 Architectural design data in AutoCAD Application's Representations
8.5.3 Architectural design data in PSL syntax (KIF Notation)
8.5.4 Architectural design data in PSL Representations
8.5.5 Architectural design data in PSL syntax and CCS terminology
8.5.6 Architectural design data in CCS Application's input File
8.5.7 Cost Estimating Output File in CCS Application's Representations
8.5.8 Cost Estimating data in PSL syntax and CCS terminology
8.5.9 CCS Cost Estimating data in PSL Representations
8.5.10 Cost-Estimating data in PSL syntax and MS project terminology
8.5.11 Cost-Estimating data in MS Project Application's Representations
8.5.12 MS project Scheduling data in PSL syntax & MS project terminology
8.5.13 MS project Scheduling data in PSL in PSL Representations

9.3.6.1 Translation Process

The translation process begun with a file, in **section 8.5.2,** that is written in the AutoCAD Application's syntax and using the application's terminology (AutoCAD application's native ontology). This native ontology is developed from the application's data types and the corresponding process related data for the case study project, which are the project's design elements (or drawing objects) and their components parts, that represent the construction or manufactured products, methods and materials of construction for the component parts. The syntactic parser reads the file (in **section 8.5.2)** and produced a corresponding file using PSL syntax (KIF notation), but still preserving the application's terminology as shown in **section 8.5.3.** Using the semantic translator (hand written semantic definitions) developed between AutoCAD and PSL the design file (of **section 8.5.3)** in KIF syntax and the application's terminology is translated to a file (in **section 8.5.4)** containing only PSL terminology and syntax by substituting the definitions of all AutoCAD application's terms with their definitions in PSL that are determined by the translation definitions developed between AutoCAD and PSL. Following a reversed step the design data file (of **section 8.5.4)** in PSL representations (PSL syntax and terminology) was translated to CCS application's representation (terminology).

Using the semantic translator (hand written semantic definitions) developed (for CCS) between CCS and PSL, the design data file (of **section 8.5.4**) in PSL representations were mapped to a file containing only CCS application's terminology to produce the file (in **section 8.5.5**) containing only CCS application's terminology but still preserving the PSL syntax. A reveres syntactic parser read the file (in **section 8.5.5**) in PSL syntax (KIF notation) and CCS application's terminology and produced a corresponding file (CCS applications native ontology) in CCS application's syntax and the application's terminology as shown in **section 8.5.6.** A cost estimator then could import this file to CCS application's native file and generate cost data for the case study project. This cost data is formatted in CCS application's report files, which have different data formats form the input, as shown by the CCS application's native output ontology given in **section 8.5.7**. The syntactic parser reads the file (in **section 8.5.7**) and produced a corresponding file using PSL syntax (KIF notation), but still preserving the CCS application's terminology as shown in **section 8.5.8.** Using the semantic definitions (hand written semantic translator) developed for CCS application the cost the data file (of **section 8.5.8**) in KIF syntax and the applications terminology was translated to a file (in **section 8.5.9**) containing only PSL terminology and syntax by substituting the definitions of all CCS application's terms with their definitions in PSL that are determined by the translation definitions developed between CCS and PSL. Using the semantic definitions (hand written semantic translator) developed for Micro soft project application, the file (of **section 8.5.9**) in PSL representations were mapped to a file containing only Microsoft project application's terminology to produce the file (in **section 8.5.10**) containing only Microsoft project application's terminology in PSL syntax (KIF notation). A reveres syntactic parser read the file (in **section 8.5.10**) in PSL syntax (KIF notation) and Microsoft project application's terminology and produced a corresponding file (Micro soft project applications native ontology) using the Microsoft project application's syntax and the application's terminology as shown in **section 8.5.11.** The syntactic parser reads the file (in **section 8.5.11**) and produced a corresponding file using PSL syntax (KIF notation), but still preserving the Microsoft project application's terminology as shown in **section 8.5.12.** Finally using the semantic definitions (hand written semantic translator) developed for Micro soft project application, the file (of **section 8.5.12**) in KIF syntax is translated to a file (in **section 8.5.13**) containing only PSL terminology and syntax by substituting the definitions of all Micro soft project application's terms with their definitions in PSL that are determined by the translation definitions developed between Micro soft project application and PSL

9.3.7 Proposal for New Extensions to PSL Ontology

From analysis of results of the scenario applications data types mapping to the PSL ontology and information translation between the scenario software applications' representations using PSL based translators (translation definitions) developed between each of the applications and PSL, it was identified that a complete representation and exchange of construction process information can not be guaranteed using the exiting PSL concepts. If PSL is to facilitate the complete representation and exchange of process information between the scenario applications, additional extensions need to be developed to the PSL ontology to incorporate construction process concepts related to the pre-construction process stages that are mew to PSL ontology. These concepts do not include every aspect of the applications but they were identified from the shared concepts between the scenario applications. Therefore the semantic concepts of the applications data types that represent shared information need to be defined in the PSL ontology new extensions. The proposed extensions including the construction process concepts, the relationships between them, and the theories required from other existing PSL extensions are presented in the following section as follows:

(a) Construction or Project Product

Extension Name: Construction or Project_Product.def
Primitive Lexicon: None

Defined Lexicon:

- (design_elements ?x)
- (components ?x1.. ?x)

Theories Required by this Extension: Processor_Activity.def, psl_core.th

(b) Cost

Extension Name: Construction or cost.def
Primitive Lexicon: None

Defined Lexicon:

- (resource_rate ?x)
- (activity_rate ?x)
- (product_rate ?x)
- (activiy_total_cost ?x)
- (activiy_fixed_cost ?x)
- (product_cost ?x)
- (resource_cost/use ?x ?a)
- (resource_standad_rate ?x ?a)
- (resource_category_rate ?x)
- (resource_category_cost ?x)
- (overhead_cost ?x)
- (mark_up ?x)
- (split_rate ?x)
- (plug_rate ?x)

Theories Required by this Extension: Processor_Activity.def, psl_core th, requires th, Project_Product def

(c) Activity and Product Quantity

Extension Name: Activity_&/or_Product _Quantity.def
Primitive Lexicon: None

Defined Lexicon:

- (activity_billed_quantity ?x)
- (product_quantity ?x)

Theories Required by this Extension: Processor_Activity.def, sub_activiy.th, psl_core.th, Project_Product def,

(d) Resource Category

Extension Name: Resource_Category.def
Primitive Lexicon: None

Defined Lexicon:

- (plant ?x)
- (labour ?x)
- (material ?x)
- (temporary_material ?x)
- (sub_contract ?x)

Theories Required by this Extension: requires.th , psl_core.th, res_divisible def

9.4 Web Services: As a Possible Alternative to the use of PSL

Several groups have been working on various aspects of standards and different approaches have been taken to achieve integration and interoperability of heterogeneous software applications in the construction industry. In the 1970s and early 1980s through the standardization of the information formats, such as IGES and later DXF. Since late 1980s conceptual models are being standardized within ISO and IAI. While these and almost all other efforts for construction software interoperability focus on product model data, to date, the process-centered data of construction projects have received little attention. However at National Institute of Standards and Technology (NIST) an effort is under way to standardize a "process description language" called Process Specification Language (PSL) for process related manufacturing applications' interoperability. This research study has investigated the use and applicability of this emerging standard process specification and exchange language in the construction environment and evaluated its impact in construction software interoperability and the results are discussed in this chapter

As an alternative to the use of PSL for process related construction software applications' interoperability web services may offer the possibility of another choice. A web service as defined by the W3C Web Services Architecture Working Group (http //www.w3 org/ws/arch) is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML-based messages via internet-based protocols.

Web services provide a standardized way of integrating web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. The functions of these open standards are:

XML, Extensible Markup Language: is used to tag the data. XML is a specification developed by the W3C XML is designed especially for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

Web services use XML-based messaging as a fundamental means of data communication to help bridge the differences that exist between systems that use different component models, operating systems, and programming languages Both the Web service client and the Web service provider are freed from needing any knowledge of each other beyond inputs, outputs and location.

SOAP, Simple Object Access Protocol: is used to transfer the data, SOAP is a lightweight XML-based messaging protocol used to encode the information in Web service request and response messages before sending them over a network. SOAP messages are independent of any operating system or protocol and may be transported using a variety of Internet protocols, including SMTP, MIME, and HTTP

***WSDL, Web Services Description Language*:** is used for describing the services available *WSDL, is* an XML-formatted language used to describe a Web service's capabilities as collections of communication endpoints capable of exchanging messages. WSDL is an integral part of UDDI, an XML-based worldwide business registry. WSDL is the language that UDDI uses

UDDI, Universal Description, Discovery and Integration: is used for listing what services are available. ***UDDI,*** a Web-based distributed directory that enables businesses to list themselves on the Internet and discover each other.

Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind. In this way the scenario software applications of this research study and other construction applications may be able to interact and exchange information over the Internet using web services. However there is still the need for a standard language to resolve the semantic differences between the applications' information representations. The facility supported by web services alone is not enough to support interoperability of construction software applications. For complete and efficient software interoperability the semantics of the information exchanged using web services need to be compromised through the use of a standard specification language. To data, communication via e-mail and document sharing via the WWW are well-established practices. Increasingly, the Internet technologies have provided the potential for distributed systems and generic data exchange. However, for these technologies, to support complete and efficient software interoperability and collaboration in construction a standard specification language is required to resolve the semantic differences between the applications information representations

The use of web services in conjunction with construction process specification language standardization will lead to richer, more industry-specific data exchange methods. This is important for the construction industry where information is complex with different representation structures and terminology or language and communication is highly important and will lead to complete AEC/FM software interoperability. Hence web services cannot be an alternative to the use of PSL in construction but as a framework that allow the construction software applications to interoperate using PSL through web services.

Chapter 10 Conclusions and Recommendations

10.1 Introduction

The preceding chapter discussed the investigation of the applicability of the Process Specification Language (PSL), which was developed for manufacturing process information specification and exchange, in the construction environments, and the implementation of PSL language in a construction scenario. This section summarises the result of the research with regard to the following questions:

How applicable is PSL in construction environment?

Provided proved applicable, what is the impact of PSL in construction data interoperability?

What is the impact of construction process concepts on the use of PSL in construction?

Then recommendations for further work are briefed.

10.2 Conclusions

PSL should be applicable in construction provided extensions are developed to incorporate process concepts for defining construction process information representations, that the current PSL ontology lack or those are new to the current PSL ontology. The mapping of the AutoCAD, CCS and Microsoft project scheduling applications data types, which are process information representations with in the pre-construction process stages, to the PSL ontology identified the PSL parts that contains applicable concepts to define those data types and the need for new extensions to the PSL ontology that would incorporate the semantic concepts of the construction project products or project design elements represented by AutoCAD drawing objects; cost, resource category, and billed quantities (quantity of CCS bill items) of CCS and cost data types of Microsoft project applications. For PSL to support the complete and correct process information exchange between software applications that support the pre-construction process stages, the semantic concepts of these applications process information representations (data types) need to be clearly and completely defined within the PSL ontology.

PSL is a promising standard language for construction process information exchange between construction software applications. The implementation of PSL in the construction scenario for the case study project construction process information exchange between the scenario applications, has shown the potential of the language to resolve the semantic differences that hampered the electronic information exchange between construction applications. This research study has identified the data types of the AutoCAD design, CCS cost estimating, and Microsoft project scheduling applications and their semantic concepts need to be defined clearly in the PSL ontology in order to render these applications PSL compliant. Once these applications become PSL compliant by defining the semantic concepts of their process information representations (or data types), they would be able exchange information with each other and with every other application that is PSL complaint.

The compliance of construction software applications to PSL in construction practices would reduce:

- The efforts of obtaining the correct information
- The risk of misinterpretation of information at the receiving/destination end
- The risk of losing information during exchange
- The time wasted in obtaining the correct information

Although this research study has focused on process information exchange between construction applications and software interoperability in the construction sector, the benefit that can be obtained from PSL would not be restricted within the industry. For example PSL would also integrate construction with the manufacturing and other PSL compliant sectors, which contribute to the success of construction practices. This would allow the construction professionals to exchange information with manufacturers and suppliers, for instance designers for manufactured product data and cost estimators and schedulers for supplier's information.

From the result of the construction process information representations within the pre-construction process stages mapping to the PSL ontology based on multidimensional conceptual analysis between the construction process concepts natural language definitions and the manufacturing process concepts definitions within the PSL ontology, it was concluded that PSL is applicable in construction provided extensions are developed to incorporate semantic definition for the construction process concepts that are new to the PSL ontology or for construction specific process concepts if such concepts are identified.

10.3 Recommendations for Further Research

PSL was designed for process information however from the implementation of PSL in the construction scenario which involved information exchange between CCS cost estimating, and Microsoft project scheduling including AutoCAD design, has proven that PSL should be able to be used for product data as well. PSL can be extended to incorporate the concepts of Product. Product is related to process definitions through the processes that are accomplished to develop the product. However PSL ontology to date has not defined the concepts of product, hence a new extension is required to incorporate the semantic concepts of product/construction product/ project product/ or project design elements.

Cost information of construction processes is the main issue in the pre-construction process stages and cost related information is exchanged between cost estimating and scheduling applications. Additionally as described in the exchange scenario example in **chapter 7**, a designer may be required to produce design information for a project to a specified budget hence the designer may need to exchange cost information with the other two applications. There are no semantic concepts for cost in general in the current PSL ontology. Hence there is a need for further extensions to the PSL ontology to incorporate the concepts of product, activity, and resources costs.

PSL is a language for specifying processes through process concepts and relationship between them using KIF. Construction applications represent process information using their data types and relationships between them. In order for an application to be PSL complaint the semantic concepts of the application's data types, which represent the data that need to be exchanged between applications, should be defined in the PSL ontology This has to be done in the same way that these applications define or represent the required process information, which is normally done through the organization and relationships of the data types in the applications. However information representation data types in different applications are organized differently to represent information. During the information exchange between the scenario applications using PSL, it was identified that some of the information has to be mapped to concepts from two different parts of PSL.

This obviously would create problems during information exchange between different applications (such as CCS and Microsoft project scheduling applications) with different organization of data types. Hence there is a need for further investigation about the relationships between PSL parts.

The ontology based approach to PLS compliance is different to the traditional approach to standards compliance that rather than forcing the adoption of exactly the same terminology, an application is PSL-compliant if there exist definitions for its terminology (semantic definitions for the application's data types) in the PSL ontology using PSL language and syntax. The questions that need further investigation are:

- If data types representing shared information between two or more applications can have different definitions in the PSL ontology; how would these applications exchange process information and be PSL compliant?. Are the translation definitions between applications and PSL the decisive construct for the applications compliance to PSL?
- In the investigation of the applicability of PSL in construction it was identified that the semantic concepts of the applications data types can be defined using existing PSL concepts however sometimes this has to be done by combing concepts from different parts of PSL. What is the effect of this in the applications compliance to PSL?. What are the relations between the parts?. How can this be resolved if it causes problem to the compliance of applications or information exchange between applications? Are the translations definition enough to resolve this problem?.

For correct and complete translation, translators (or translation definitions) must be based on the *formal* specifications of the representation's semantics (the data types semantic). In the implementation of PSL in the construction scenario the information translation between the scenario applications' representations was carried out using hand written translators i.e. information was exchanged between the applications by hand using parsing procedures for syntax translation, and mapping and pattern matching of concepts using the PSL based hand written semantic translation definitions for semantic translation.

The Development of electronic translation programs would provide better basis for understanding:

- The impact of PSL as interchange language in construction practices
- The effect, of the use of concepts from different parts of PSL for defining a shared data, on the compliance of applications to PSL and information exchange between applications.
- The roles of translations definitions. Can the translation definitions specify the relationships between the PSL parts? How can the translation definitions resolve the problem of information exchange that may arise from the use of different PSL parts for a shared data?
- The impact of construction process concepts on the use of PSL as interchange language for process information exchange between construction applications.

After the initial development of the PSL ontology from a single scenario, multiple pilot implementations of PSL in the manufacturing environment were planned to further extend the ontology From the literature it was identified that there was only one implementation (Schlenoff, C., et. al 1999) conducted when PSL was used to integrate two manufacturing applications; the ProCAP (KNO98) process modeling and the ILOG scheduler (ILO98) packages. However, the PSL ontology has been extended further after this implementation. Were these extensions developed based on PSL implementation on scenarios or are they speculative extensions for possible future use by applications? The problem is that different applications have different information representations or data types organized very differently, hence data types relate differently to describe processes or represent process information. Therefore, if there is a plan to further extend PSL ontology to incorporate construction process concepts and make PSL a standard language in the construction environment, it is recommended that the PSL ontology is extended often on implementation or trials of the language in scenarios that involve information exchange between construction applications

References

Amor R W and Hosking J G (1993). *Multi-disciplinary views for integrated and concurrent design*, in Management of Information Technology for Construction, K S Mathur, M P Betts and K W Tham (eds), World Scientific, Singapore, pp.255-267.

Ang, C. L., 1999, *Enactment of IDEF0 Modes.* International Journal of Production Research, to be published.

Argyris, Chris; Putnam, Robert and Smith, Diana McLain, *Action science: concepts, methods and skills for research and intervention.* San Francisco, Ca : Jossey-Bass., 1985.

Ausubel D.P., *Educazione e processi cognitivi*, Franco Angeli, 1968.

AutoCAD Design applications, AutoDesk product

Avison, D. Lau, F. Myers, M. and Nielsen, P. A. " *Actions research*" Communications of the ACM, Vol. 42, 19999, pp. 94-97

Avison, D., & Fitzgerald, G.; and Wood-Harper, A. "Information Systems Development: A Tool Kit Is Not Enough," *The Computer Journal* (31.4), 1988, pp. 379-380.

Avison, D., & Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools*, Blackwell Scientific Publications, Oxford, 1995.

B. Curtis, M.I. Kellner, J. Over, "*Process Modelling*", in: Communications of the ACM, 35(9), 1992

B.C. Björk (1994), "Conceptual Models of Product, Project and Document Data; Essential Ingredients of CIC," *Computing in Civil Engineering· Proceedings of the First Congress*, ASCE, Vol 1, pp. 980-987.

Baskerville, R. & Pries-Heje, J. (1999) *Grounded action research: a method for understanding IT in practice.* Accounting Management and Information Technology, **9**, 1-23.

Baskerville, R. and A. T. Wood-Harper. (1998) "*Diversity in Information Systems Action Research Methods,*" *European Journal of Information Systems*, (7) 2, pp. 90-107.

Baskerville, R.L. & Wood-Harper, A.T. (1996) A critical perspective on action research as a method for information systems research. Journal of Information Technology, **11** (3), 235-246.

Benbasat, I., Goldstein, D K. and Mead, M. "*The Case Research Strategy in Studies of Information Systems,*" MIS Quarterly (11:3) 1987, pp. 369-386.

Benjamin, P. C., R. J. Mayer, T. M. Blinn, C. Marshall, and R. Tye. "*Toward an Environment for Collaborative Enterprise Engineering.*" Society for Enterprise Engineering (SEE) Conference, Orlando, FL, 1995.

Benjamin, P.C , Menzel, C.P , Mayer, R.J. and Padmanaban, N. (1995) Towards a method for acquiring CIM ontologies. International Journal of computer integrated Manufacturing 8(3), 225-34

Benveniste A., and Berry B.. *The synchronous approach to reactive and real-time systems*. Proceedings of the IEEE, 79(9):1270-1282, September 1991.

Bickman, L. (Ed.). (1986). Using program theory in evaluation. New Directions for Program Evaluation. San Francisco, CA: Jossey-Bass.

Bleicher, J. *Contemporary Hermeneutics: Hermeneutics as Method, Philosophy and Critique,* London and Boston: Routledge & Kegan Paul, 1980.

Boland, R. "Phenomenology: A Preferred Approach to Research in Information Systems," in *Research Methods in Information Systems*, E. Mumford, R.A. Hirschheim, G. Fitzgerald, and T. WoodHarper (eds.), NorthHolland, Amsterdam, 1985, pp. 193201.

Bravoco, R. R. and Yadav, S. B., 1985a, *Requirement definition architecture-an overview*. Computers in Industry, 6, 237-251.

Bravoco, R. R and Yadav, S. B., 1985b, *A methodology to model the functional structure of an organisation*, Computers in Industry, 6, 345-361

Building construction Core Model, Part106; http //huizen dds.nl/~tnobouw/

Building construction Core Model, Part106; http://www bre co.uk/~itra/ceic htm

Burrell, G and Morgan, G. *Sociological Paradigms and Organisational Analysis*, Heinemann, London, 1979.

Burstein, F. and Gregor, S. (1999). *The systems development or engineering approach to research in Information Systems*: An action research perspective. In B. Hope and P. Yoong (Eds) Proceedings of the 10th Australasian Conference on Information Systems, (pp. 122-134).Victoria University of Wellington, New Zealand, 122-134.

Bussler, C. (1994) *Enterprise Process Modelling and Enactment in GERAM, Proc.* Third Int. conf. on Automation, Robotics and computer Vision (ICARCV '94), Singapore, 8-11 November, pp. 317-21

Carr, W. and Kemmis, S. *Becoming Critical: Education, Knowledge and Action Research*, Falmer Press, London, 1986.

Carter K. and Fortune C., 2002, "*Sustainability: Explorations in Research Approaches*" Department of Building, Engineering and Surveying, Heriot-Watt University, Edinburgh, UK, EH14 4AS

Catron, B., Ray, S., ALPS: *A Language for Process Specification*, Int. J. Computer Integrated Manufacturing, 1991, Vol. 4, No. 2, 105-113.

CCS, Construction Computer Software Cost Estimating and Evaluations Application for Construction Industry, http //www.ccsuk.com

Chen, H.T. and Rossi, P.H. (1983). Evaluating with sense: The theory-driven approach. Evaluation Review. 7, 283-302.

Chen, H.T. and Rossi, P H (1987). The theory-driven approach to validity. Evaluation and Program Planning. 10, 95-103.

Chenitz, W. C., & Swanson, J. M. (1986) *From practice to grounded theory: Qualitative research.* Menlo Park, CA: Addsion-Wesley.

Clark, P.A. *Action Research and Organizational Change*, Harper and Row, London, 1972

Clayton, M.J., Fischer, M.A., and Kunz, J.C. (1998) . CAD Prototype Testing: Worked Examples, Demonstrations, Trials, and Charrettes." Computing in Civil Engineering, Proceedings of International Computing Congress, Boston, October 18-21, Kelvin C.P. Wang (Ed), ASCE, 106-116.

Colquhoun, G.J., R.W. Baines, and R. Crossley. *"A State of the Art Review of IDEF0."* International Journal of Computer Integrated Manufacturing 6, no. 4 (1993). 252-264.

COMBINE; http //erg.ucd ie/combine html

Cook, T. D , & Campbell, D. T. (1979). *Quasi-Experimentation: Design & analysis issues for field settings.* Boston, MA: Houghton Mifflin Co.

Cordray, D.S (1986). *Quasi-experimental analysis: A mixture of methods and judgment.* In W. Trochim. (Ed.). Advances in quasi-experimental design and analysis. New Directions in Program Evaluation. San Francisco, CA: Jossey-Bass.

Creswell, J. W. (1994). *Research design: Qualitative & quantitative approaches.* Thousand Oaks, CA: Sage Publications.

Curtis, Bill, M. I. Kellner and J. Over, "*Process Modeling*", Communications of the ACM, 35(9), Sept. 1992, pp. 75-90.

Cutting-Decelle A.F., Michel J,J., Schlenoff C. 2000. *From Manufacturing to construction: to wards a common representation of process information*: the PSL approach, concurrent Engineering Conference., Lyon.

D A. Marca and C. L. McGowan. *SADT : Structured Analysis and Design Techniques.* McGraw-Hill, New York, 1988.

D. T. Ross. "*Applications and Extensions of SADT.*" *Computer.* April·pp. 25-34, 1985.

David, R. & Alla, H., *Autonomous And Timed Continous Petri Nets.* Applications and Theory of Petri Nets 1991: 71-90

Davies, L.J. "*Researching the Organisational Culture Contexts of Information Systems Strategy*", in Nissen, H E., Klein, H K. and Hirschheim R. (eds.), *Information Systems Research in the 1990's.* Amsterdam, Elsevier/North Holland, 1991.

Davies, L J and Nielsen, S. "*An Ethnographic Study of Configuration Management and Documentation Practices in an Information Technology Centre*", in Kendall, K.E , Lyytinen, K. and De Gross, J. (eds.), *The Impact of Computer Supported Technology on Information Systems Development.* Amsterdam, Elsevier/North Holland, 1992.

Davis, R. Shrobe, H. and Szolovits P., *What is a Knowledge Representation? AI Magazine*, 14(1):17-33, 1993.

De Vries, B. 1995. "*Message development in the Building process*". in: Modeling of Buildings through their Life-cycle. Proceedings of the CIB W78 workshop 1995 (ed's: Fischer, Law, and Luiten) pp. 467-479, Stanford (CA): Stanford University.

Desrochers, A.A. and Al'Jaar, R. Y , *Applications of Petri Nets in Manufacturing Systems. Modelling, Control and Performance Analysis*, IEEE Press, Piscataway, USA, 1995.

Dey, I. *Grounding grounded theory: guidelines for qualitative inquiry*, Academic Press, San Diego, 1999.

Disa (1995)" *BPR Survey Summary Report*" (defense Information Systems Agency).

Dunn, W. (1981). *Public policy analysis*: An introduction. Englewood Cliffs, NJ: Prentice Hall.

Eastman 1993, *life cycle requirements for building product models*, in management of technology for construction, ed, Mathure et al. Singapore world publ. Corp.

Eastman, C.M. 1992 *modeling of buildings*: evolution and concepts. Automation in Construction , Vol.1 (2): pp.99-121

Edum-Fotwe, F.T.; Price, A.D.F. and Thorpe, A., 1996, *Research Method versus methodology: Achieving quality in scholarly research for construction management*, Proceedings of the 12th Annual Conference of ARCOM, pp 428-437

Egham, *Proceedings of the Software Process Workshop*, Egham, Surrey, UK, February 1984, edited by Colin Potts, IEEE Press.

Einhorn, H.J. and Hogarth, R.M. (1986). *Judging probable cause.* Psychological Bulletin. 99, 1, 3-19.

Ekholm A., *A conceptual framework for classification of construction works*, ITCON Vol. 2, 1996

Elden, M. and Chisholm, R.F. "*Emerging Varieties of Action Research: Introduction to the Special Issue*," Human Relations *(46:2)*, 1993, pp. 121-142.

ESPRIT ATLAS; http://omega sop cstb fr/ILC/ecprojects/atlas/atlas.html

F. Vernadat. CIMOSA - *A European development for enterprise integration. Part 2: Enterprise modelling.* In Enterprise Integration Modeling (C. Petrie, Ed) The MIT Press, Cambridge, MA. 1992. pp. 189-204.

Fellows, R and Liu, A., 1997, *Research Methods for Construction book*, Blackwells, Oxford

Ferscha, A , 1998, "*Optimistic Distributed Execution of Business Process Models*," *Proceedings of 31st Annual Hawaii International Conference on System Sciences*, 723–732.

Firestone, W. "*Accommodation· Towards a Paradigm-praxis Dialectic*," in E. Guba (ed) *The Paradigm Dialog*, Sage, Thousand Oaks, CA, 1990, pp. 105-124.

Fitzgerald, B. & Howcroft, D. 1998: Competing dichotomies in IS research and possible strategies for resolution ICIS 1998: 155-164

Fitzgerald, B. and Howcroft, D. "Toward dissolution of the IS research debate from polarization to polarity," *Journal of Information Technology* (13), pp. 313-326. Beyond the Approach of Positivism Alone," *MIS Quarterly* (23:1), March 1999, pp 29-34.

Fitzgerald, B. and Murphy, C.1994), "the new Industrial Engineering information technology and business process redesign" Sloan Management Review, Summer, p11-27.

Fowler, J. (1996). *Overview of STEP Architecure and AP Development.* Presentation to STEP Building Construction Committee, Dallas, January 1996.

Fox, M., et al, *An Organization Ontology for Enterprise Modeling: Preliminary Concepts",* Computers in Industry, 1996, Vol. 19, pp. 123-134.

Froese T (1994), A Summary of: Workshop On The Standardization Of Information Structures In The Construction Industry

Froese T. (1994b), "*Developments to the IRMA Model of AEC Projects," Computing in Civil Engineering: Proceedings of the First Congress,* ASCE, Vol. 1, pp. 778-785.

Froese T. and K. Q. Yu, "*Industry Foundation Class Modeling For Estimating And Scheduling",* Durability Of Building Materials And Components 8. Vancouver, May 1999. Vol. 4, pp. 2825-2835.

Froese T., "*Information Standards In The AEC Industry," Canadian Civil Engineer,* Vol. 11, No. 6, September 1994, pp. 3-5.

Froese T., Grobler F., and Yu K., "*Development of Data Standards for Construction--An IAI Perspective*," B-C. Björk and A. Jagbeck (Eds.), The Life-Cycle of Construction IT Innovations-Technology Transfer from research to practice, Proceedings of the W78 conference, Stockholm, Sweden, June 3-5, 1998. KTH, Stockholm, pp.233-244.

Gable, C. "*Integrating Case Study and Survey Research Methods*: An Eexample in Information Systems," *European Journal of Information Systems* (3:2), 1994, pp. 112-126.

Gadamer, H-G. "*The Historicity of Understanding*," in Critical Sociology, Selected Readings, P. Connerton (ed.), Penguin Books Ltd, Harmondsworth, 1976b, pp. 117-133.

Galliers, R. (1991) "*Choosing appropriate information systems research approaches: a revised taxonomy*" in H. Nissen, H. Klein and R. Hirschheim, (eds.) *Information Systems Research: Contemporary Approaches & Emergent Traditions,* North-Holland. Elsevier Science Publishers, 327-345.

Genesereth, M. and Fikes, R. (1992). *Knowledge Interchange Format, Version 3 0,* Computer Science Department, Stanford University edition.

Genesereth, M. R., and Fikes, R. E. (1992). *Knowledge Interchange Format,* Version 3.0 Reference Manual. Technical Report Logic 92-1, Computer Science Department, Stanford University.

Glaser, G. G., & Strauss, A. L. (1967). *The discovery of grounded theory: Strategies for qualitative research.* NY: Aldine De Gruyter.

Gomez-Perez, A. 1998. "*Knowledge Sharing and Re-Use.*" In *The Handbook of Applied Expert Systems,* J. Liebowitz, ed. Boca Raton, 10:1-10:36.

Grady Jeffrey O., *System Requirements Analysis,* New York: McGraw-Hill, 1993.

Gruber T., (1991) Ontologua: A Mechanism to Support Portable Ontology. Research report KSL-91-66, Stanford Knowledge Systems Laboratory, Stanford University, Stanford, CA.

Gruninger, M. and Fox, M>S. (1994) An activity ontology for enterprise modelling. Proc. Third IEEE Workshop o Enabling Technologies; Infrastructures for Collaborative Enterprise (WET ICE 94), Morgantown, West Virginia.

Gruninger, M., Sriram, R.D., Cheng, J., Law, K., "*Process Specification Language for Project Information Exchange,*" International Journal of IT in Architecture, Engineering & Construction, 2003.

Guba, E. (Editor). *The Paradigm Dialog,* Sage, Thousand Oaks, CA, 1990.

Guba, E. and Lincoln, Y. (1994) *Competing paradigms in qualitative research,* in Denzin, N. and Lincoln, Y. (Eds.) (1994) *The Handbook of Qualitative Research,* Sage Publications, California, 105-117.

Hayes, P., *A Catalog of Temporal Theories,* Technical Report UIUC-BI-AI-96-01, TheBeckman Institute, University of Illinois, 1996.

Hoare 1985, Prentice-Hall, Englewood Cliffs NJ.– Doc 321 A Calculus of Communicating Systems - Milner - 1980 Doc 105 Theoretical Computer Science 92, Springer verlag, Berline, Doc 1864 Communicating Sequential Processes

Hoare, C.A.R (1985), *Communicating sequential processes,* Prentice Hall, London.

Holzblatt, K. and Beyer, H. "*Making Customer-Centered Design Work for Teams*", *Communications of the ACM,* (36.10), 1993, pp 93-103.

Hughes, J.A., Randall, D. and Shapiro, D. "*Faltering from Ethnography to Design*", *CSCW '92*. ACM 1992 Conference on Computer-Supported Cooperative Work: Sharing Perspectives. New York, ACM Press, 1992, pp. 115-123.

Hurricane Engineering and Testing INC.,
https://secure.metaldoor.net/dominionproducts/specifications/specs110.cfm

IAI (1996), "*End User Guide to Industry Foundation Classes, Enabling Interoperability in the AEC/FM Industry*". International Alliance for Interoperability (IAI), 1996.

IAI (1997a), "*IFC Object Model For AEC Projects, Specifications Volume 2*", IFC R1.5 Final, 1997.

IAI (1997b), "*FM Project, FM-4: Occupancy Planning*", proposal for IFC R2.0, Facilities Management Domain, International Alliance for Interoperability (IAI), 1997

IAI (1997c), "*Guidelines for the Development of Industry Foundation Classes*" draft, International Alliance for Interoperability (IAI), May, 1997

IAI, 1997, *IFCs Object Model*, Release 1.5, In: CD_ROM, Industry Foundations classes Release 1.5, IAI members CD, International Alliance for Interoperability.

IAI, IFC release 2x, IFC technical Guide (TG), 2000

IAI, Transposing the IFC2x Specification as ISO Standard, 2001

IAI. (1996). Industry Alliance for Interoperability. http://www.interoperability.com/

ICAM. *Integrated computer-aided manufacturing (ICAM), Architecture* Part II. Volume IV -Function modelling manual (IDEF0). Waltham, MA, US, SofTech, 1981, 165 p.

IDEF0 1993, Integrated Definition for Function Modelling Draft Federal Information Processing Standards Publication183

IDEF0, 1981, "ICAM Architecture Part II, Vol. 5," *Information Modeling Manual (IDEF0)*, Report number AFWAL-TR-81-4023, Vol. 5 Mantech Technology Transfer, Center WL/MTX.

IDEF0. *Integration definition for function modeling (IDEF0).* Federal Information Processing Standards publication 183, 1993.

Iivari, J. "*A Paradigmatic Analysis of Contemporary Schools of IS Development*," *European Journal of Information Systems* (1:1), 1991, pp. 249-272.

ILO98, "ILOG Scheduler 4.3 Reference Manual," June 1998.

Information Integration for Concurrent Engineering (IICE, 1994) IDEF5 method Report, Knowledge based systems , Inc. Armstrong Laboratory AL/HRGA, wright-Patterson Air force Base, Ohio 45433.

Information Integration for Concurrent Engineering (IICE, 1995) IDEF3 method Report, Knowledge based systems , Inc.

ISO (1994 a). *Classification of information in the construction industry.* ISO Technical Report 14177:1994(E). Geneva: International Organisation for Standardisation.

ISO (1995). *Product Data Representation and Exchange*, Part 225. Building Elements using Explicit Shape Representation, Committee Draft, ISO TC184/SC4/WG3 N434.

ISO (2001) ISO CD 18629-1, *Industrial automation systems and integration – Process specification language*: Part 1. Overview and basic principles, Geneva: ISO

ISO 10303-21:1994, *Industrial automation systems and integration – Product data representation and exchange* – Part 21: Implementation methods: "Clear text encoding of the exchange structure.

ISO 10303-23(DIS) , *Industrial automation systems and integration – Product data representation and exchange* – Part 23. Implementation method: C++ language binding to the standard data access interface.

ISO CD 18629-41, *Industrial automation system and integration – Process specification language*: Part 41: Definitional extension: activity extensions, ISO TC184/SC4/JWG8 N335, 2003

ISO DIS 18629-1, *Industrial automation system and integration – Process specification language* . Part 1 overview and basic principles, ISO TC184/SC4/JWG8 N324, 2003

ISO DIS 18629-12, *Industrial automation system and integration – Process specification language* · Part 12 : Outer Core, ISO TC184/SC4/JWG8 N349, 2003

ISO DIS10303 Part 225, *Building Elements Using Explicit Shape Representation* ICS 25 040 40

ISO IS 10303-1, "*Industrial automation systems and integration – product data representation and exchange – part 1 : overview and fundamental principles*", 1994

ISO TC 10/ SC8/ WG13. *Computer-Aided Design (CAD) Technique* - Use of computers for the preparation of construction documentation, Technical note concerning the scope of the working group, 1993.

ISO TC 59/SC13. *Classification of Information in the Construction Process*, ISO working group TC59/ SC13 WG2, Technical Report, December 1993

ISO TC10/SC8/WG13, 1996, *Implementation of ISO13567 CAD Layer Standard*, (DRAFT) World Wide Web http://www.ce.kth.se/fba/bit/cadlayer.htm

ISO WD 10303-Part 22, 1993, *Standard Data Access Interface*, SOLIS Repository, NIST, Gaithersburg, MD

ISO, 1998b, *Technical Product Documentation, Organisation and naming of layers for CAD*, part 1: overview and principles, part 2: concepts, format and codes used in construction documentations ISO 13567:1998.

ISO. (1994a). ISO-10303-11: *Project Data Representation And Exchange* Part 11: Description Methods, The EXPRESS Language Reference Manual, ISO/ IEC

ISO. (1994b). ISO-10303-21: *Project Data Representation And Exchange* Part 21: Implementation Methods, Clear Text Encoding Of The Exchange Structure, ISO/

ISO/TC184/SC4, 1993a, *The STEP Standard, Draft International Standards* ISO/DIS 10303 (to be published in several parts)ISO Technical committee 184.

ISO10303-24 (CD), *Industrial automation systems and integration* – Product data representation and exchange – Part 24: Implementation method: C language binding to the standard data access interface.

ISO10303-26 (CD), *Industrial automation systems and integration* – Product data representation and exchange – Part 26: Implementation method: Interface definition language binding to the standard data access interface.

IT in Civil Engineering 2003, *Joint Symposium on Information Technology in Civil Engineering*, Information Technology Symposium- journal, November 12-15, 2003

Jagbeck A. 1998 , *IT Support for Construction Planning. A system design based on integrated information* (Thesis). Royal Institute of Technology, Construction management and economics, Stockholm, ISBN 91-7170-269-5.

Jick, T "*Mixing Qualitative and Quantitative Methods*: Triangulation in Action," in J. Van Maanen (ed.), *Qualitativ Methodology*, Sage, Thousand Oaks, CA, 1983, pp. 135-148.

Jnell, http://www nist gov/sc5/soap/

Johnson RB (1997) *Examining the validity structure of qualitative research.* Education, 118(2): 282 –292.

Jos P. Van Leeiwem, 1999, *Modeling Architectural Design Information by features.* Research project thesis.

Kamath, M., and Viswanadham, N., 1986, "*Applications of Petri Net Based Models in the Modeling and Analysis of Flexible Manufacturing Systems*," in Modeling and Analysis of Flexible Manufacturing Systems, A. Desrochers, (eds), IEEE, 262–267.

Kaplan, B. and Maxwell, J A. (1994) "*Qualitative research methods for evaluating computer information systems*" in J.G. Anderson, C.E. Aydin and S.J. Jay (eds.), *Evaluating Health Care Information Systems: Methods and Applications*, Sage, Thousand Oaks, CA, 45-68

Kaplan, B., and Duchon, D "*Combining Qualitative and Quantitative Methods in IS Research.* A Case Study," *MIS Quarterly* (12.4), 1988, pp.571-587.

KBSI, *The IDEF5 Ontology Description Capture Method Overview.* KBSI Report 1994,Texas.

Kemmerer, S. J., *The grand experience*, NIST SP 939, 1999

KNO98, Knowledge Based Systems, Inc., "ProCAP Automated Process Modeling for Windows User's Manual," 1998.

Knowledge Based Systems, Inc. (KBSI), 2000, "*IDEF5 Ontology Description Capture Overview*", IDEF5 Overview, Online Available. http://www.idef.com/idef5.html

Knutilla, A., et. al , "*Process Specification Language: Analysis of Existing Representations*," NISTIR 6133, National Institute of Standards and Technology, Gaithersburg, MD (1998).

Kommunikation mit Automaten. Petri, C.A., Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962, Second Edition:, New York: Griffiss Air Force Base, Technical Report RADC-TR-65--377, Vol.1, 1966, Pages: Suppl. 1, English translation.

Krippendorff, K. *Content analysis : An introduction to its methodology.* Sage Publications, Beverly Hills, 1980.

Kuhn, T S. (1970). *Logic of discovery or psychology of research*; In I. Lakatos & A. Musgrave (Eds.), *Criticism and the growth of knowledge* (pp. 1-23). New York: Cambridge University Press.

Landry, M., and Banville, C. "*A Disciplined Methodological Pluralism for MIS Research*," *Accounting, Management and Information Technologies* (2:2), 1992, pp. 77-98

Landry, M., and Banville, C. "A Disciplined Methodological Pluralism for MIS Research," *Accounting, Management and Information Technologies* (2:2), 1992, pp. 77-98.

Larry Whitman, Brian L. Huff, Adrien Presley: *Structured Models and Dynamic Systems Analysis:* The Integration of the IDEF0/IDEF3 Modeling Methods and Discrete Event Simulation. Winter Simulation Conference 1997: 518-524

Lee, A. "*Integrating Positivist and Interpretivist Approaches to Organizational Research*," *Organizational Science* (2:4), November, 1991, pp. 342-365.

Lee, A. S , Liebenau, J. and DeGross, J. I. (Eds) (1997), *Information Systems and Qualitative Research,* Chapman and Hall, London.

Lee, J. (ed.), Gruninger, M., Jin, Y., Malone, T., Tate, A., Yost. G , and other members of the PIF Working Group, *Process Interchange Format and Framework*, Version 1.1, MIT Center for Coordination Science, Working Paper No. 194, 1996. http://ccs.mit.edu/pif/

Lee, J., et al, "The PIF Process Interchange Format and Framework", The Knowledge Engineering Review, Vol. 13(1), pp. 91-120, Special Issue on "Putting Ontologies to Use" (eds. Uschold, M. and Tate, A.), Cambridge University Press.

Lewis, I.M. *Social Anthropology in Perspective.* Cambridge University Press, Cambridge, 1985.

Luiten G. (1994), *Computer Aided Design for Construction in the Building Industry,* The Hague, Netherlands

M. Kamath and N Viswanadham, "*Application of Petri based models in the modeling and analysis of flexible manufacturing systems,*" in <i>Proc. IEEE Conf Robotics Automat.</i>, San Francisco, CA, Apr 1986, pp. 312-317.

Ma Zhiliang. "Life Cycle Support for AEC Product and Process through Computeraisaion", center for integrated research in science and engineering. Nagoya University. Nagoya, Japan.

Ma, Z. (1998) "*Life Cycle Support for AEC Product/Process through Computerization*": Department of Civil Engineering Tsinghua University, Beijing, P. R. China Visiting Scholar, Center for Integrated Research in Science and Engineering, Nagoya University, Nagoya, Japan

Marca, D.A & McGowan, C. L. (1986], "SADT: *Structured Analysis and Design Techniques*": New York, .U.S McGraw-Hill Publishing Company., Inc.

Marca, D.A and McGowan, C.L. (1988), " SADT: Structural Analysis and Design Technique": New Yourk, NY, McGraw-Hill Book Co , Inc./Snodgrass, B.N, (1993), " Integrated Activity Based Costing with IDEF modeling". Dalas

Martin, P.Y. and B.A. Turner. "*Grounded Theory and Organizational Research,*" *The Journal of Applied Behavioral Science,* (22:2), 1986, pp. 141-157.

Mathiassen, L. (1997) *Reflective Systems Development,* Vol. 1 Department of Computer Science, Aalborg University, Aalborg.

Mayer R. J., Christopher P. Menzel, Michael K. Painter, Paula S. deWitte, Thomas Blinn, and Benjamin Perakath. Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report. Technical report, Knowledge Based Systems Inc., KBSI-IICE-90-STR-01-0592-02, 1995.

Mayer, R. J. (1992). *IDEF0 Function Modeling* - A Reconstruction of the Original Air Force Wright Aeronautical Laboratory Technical Report - AFWAL-TR-81-4023 (the IDEF0 Yellow Book). College Station, Texas, Knowledge-Based Systems

Mayer, R. J., M. Painter and P. DeWitte. 1992. *IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications.* College Station, TX: Knowledge Based Systems, Inc.

Mayer, R. J., T. P. Cullinane, et al (1992). Information Integration For Concurrent Engineering (IICE) - *IDEF3 Process Description Capture Method* Report, Knowledge Based Systems, Inc.

Mayer, R.J., Benjamin, P.C., Caraway, B E. and Painter, M K (1995) *A Framework and a Suite of Methods for Business Process Reengineering.* In Grover, V. and Kettinger, W.J. (Eds.), Business Process Change: Concepts, Methods and Technologies, Idea Group Publishing, Harrisburg, PA, pp. 245-290.

Mayer, R.J., Cullinane, T.P., deWitte, P.S., Knappenberger, W.B., Perakath, B., SueWells, M., Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report, AL-TR-1992-0057, Air Force Systems Command, Wright-Patterson Air ForceBase, Ohio, USA, 1992.

McCaffer, R. and Edum-Fotwe, F., 1999, *Moving on from the crossroads: an agenda for research development in construction management*, Proceedings of the 15th Annual Conference of ARCOM, Liverpool John Moores University, pp 3-11

McGrath, J. *Groups· Interaction and Performance*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

Menzel. C , Mayer. R.J. & Edwards. D. (1990), "*IDEF Process Description and their Semantics* " Kusiak, A , and Dagli, C., (eds). Knowledge Base Systems in Design and Manufacturing, Chapman Publishing.

MetaSoftware. 1997. *WorkFlow modeler user's manual for MS Windows*. MetaSoftware Corporation: Cambridge, MA.

Microsoft (1998), Microsoft Project scheduling application, http //www mvps org/project

Miles, M. B. and A. M. Huberman (1994). Qualitative Data Analysis: An Expanded Sourcebook. Thousand Oaks, CA, Sage Publications, Inc.

Miles, M , and Huberman, A. *Qualitative Data Analysis: A Sourcebook of New Methods*, Sage, Beverley Hills, CA, 1984. Olerup, A. "Design Approaches: A Comparative Study of Information System Design and Architectural Design," *The Computer Journal* (34:3), 1991, pp. 215-224.

Miles, M B. and Huberman, A.M. 1993. *Qualitative Data Analysis: a Sourcebook for New Methods*. Newbury Park, CA.: Sage.

Milner R. (1980), *A calculus of communication systems*, LNCS 92, Springer-Verlag.

Morey, N., and Luthans, F. "*An Emic Perspective and Ethnoscience Methods for Organizational Research,*" *Academy of Management Review* (9:1), 1984, pp. 27-36.

Murata, T., ``*Petri nets: properties, analysis and applications*''; Proceedings of IEEE, vol.77, no.4, pp.541-580, 1989.

Myers, M. D., 1999. *Investigating Information Systems with Ethnographic Research* Communications of the AIS 2 (23), 1-20.

N Viswanadham and Y. Narahari. *Performance Modeling of Automated Manufacturing Systems*. Prentice Hall, Englewood Cliffs, NJ, 1992.

Nathan, B., and J Wood. 1991. *The use of IDEF0 to document a methodology - A novice's point of view*. Automation & Robotics Research Institute: Fort Worth, Texas.

Newman, W. L. (1994) *Social Research Methods: Qualitative and Quantitative Approaches*, 2nd edn, Allyn and Bacon, Needham Heights, MA.

Nijssen G M. and Halpin T.A (1989). *Conceptual schema and relational database design*. Sydney: Prentice Hall.

Nissen, H.; Klein, H.; and Hirschheim, R. (Editors). *Information Systems Research: Contemporary Approaches and Emergent Traditions*, North-Holland, Amsterdam, 1991.

Novak J.D e Gowin D.B., *Imparando a imparare*, 1989.

Nunamaker, J., Chen, M. & Purdin, T. (1991) *Systems development in information systems research.* Journal of Management Information System, 7 (3), 89-106.

Nunamaker, J., Chen, M. & Purdin, T. (1991) Systems development in information systems research. *Journal of Management Information System*, 7 (3), 89-106

Nunamaker, J , Chen, M. and Purdin, T. (1990-91) Systems development in information systems research, *Journal of Management Information Systems*, 7 (3), 89-106.

Orlikowski, W. And Baroudi, J. (1991) Studying information technology in organizations: research approaches and assumptions, *Information Systems Research*, (2:1), pp. 1-28.

Orlikowski, W.J. "*Integrated Information Environment or Matrix of Control; The Contradictory Implications of Information Technology,*" *Accounting, Management and Information Technologies* (1:1), 1991, pp. 9-42.

Patton, M. *Qualitative Evaluation and Research Methods*, 2nd ed., Sage Publications, London, 1990.

Pease, A., *Core Plan Representation* (CPR), http://www.teknowledge.com/CPR2/, November 13, 1998.

Peterson, J., Petri Net Theory and Modeling Systems, Prentice Hall, NJ, 1981.

Pettigrew, A.M. "*Contextualist Research and the Study of Organizational Change Processes*", in Mumford, E., Hirschheim, R., Fitzgerald, G. and Wood-Harper, A.T. (eds.), *Research Methods in Informations*. Amsterdam, North Holland, 1985, pp. 53-78.

PIM 1993, *Product Introduction Management*, 1993, Internal document, Lucas Industries Plc, pp1-60.

Polyak S. T., and Tate A., "*Analysis of Candidate PSL Process/Plan Representations*", AIAI-PR-66, Artificial Intelligence Applications Institute (AIAI), 80 South Bridge, EH1 1HN, Edinburgh, United Kingdom, 1997.

Polyak, S.T., A *Supply Chain Process Interoperability Demonstration Using the ProcessInterchange Format (PIF),* Report Number 889, Department of Artificial Intelligence, University of

Pouchard L. collaborative Technologies Cente Oak Ride, National Laboratories) is presented below. Web address; http://www.cs.utk.edu/~pouchard/about.html

Preston, A M. "The *'Problem' in and of Management Information Systems*", *Accounting, Management and Information Technologies*, (1:1), 1991, pp. 43-69.

PSL Ontology – Current Theories and Extensions, http://www.nist.gov/psl/psl-ontology/, June 28, 1999.

PSL, Process specification Language; http://www.mel.nist.gov/psl

Radnitzky, G. *Contemporary Schools of Metascience*, Goteborg: Scandinavian University Books, 1970.

Raftery, J.; McGeorge, D. and Walters, M., 1997, *Breaking up methodological monopolies: a multi-paradigm approach to construction management research*, Construction Management and Economics 15, pp 291-297

Raftery, J.; McGeorge, D. and Walters, M., 1997, *Breaking up methodological monopolies: a multi-paradigm approach to construction management research*, Construction Management and Economics 15, pp 291-297

RATAS. (1996a). http //www vtt fi/cic/projects/ratasimp/

RATAS. (1996b). http·//www.vtt fi/cic/projects/ratasimp/edge.htm

Reiter, R., *The frame problem in the situation calculus*: a simple solution (sometimes) and a completeness result for goal regression, Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, 1991, pages 418-440, Academic Press, San Diego.

Ricoeur, P. *The Conflict of Interpretations: Essays in Hermeneutics,* Northwestern University Press, Evanston, 1974.

Robert. P. Hanrahan, Software Technology Support Center The IDEF Process Modeling methodology.

Rogers, Whitman & Underdown, *the enterprise integration issues encouraged with agile process introduction* appeared in Flexible Automation and Integrated Manufacturing Conference Proceedings 1998 Begell House, Inc. 79 Madison Avenue, New York, NY 10016

Rosenman M A and Gero J S., *Modelling multiple views of design objects in a collaborative CAD environment* . Computer Aided Design, Vol 28(3), pages 193- 205, Elsevier science Ltd, 1996.

Ross D, 1977, "Structured Analysis and Requirements Definition," *IEEE Transactions of Software Engineering* 3 (1) pp 6-34

Ross, D.T. "Structured Analysis (SA): *A Language for Communicating Ideas,"* IEEE Transactions on Software Engineering, vol. SE-3, No. 1, pp. 16-34, 1977.

Rumbaugh, J., M. Blaha, et al. (1991). Object-Oriented Modeling and Design. Englewood Cliffs, NJ, Prentice-Hall.

Runeson, G., 1997, *The role of theory in construction management research: comment*, Construction Management and Economics 15, pp 299-302

Runeson, G., 1997, *The role of theory in construction management research: comment*, Construction Management and Economics 15, pp 299-302

Sagoo, J S. and Boardman, J.T., 1998, *"Towards the Formalisation of Soft Systems Models using Petri Net Theory," IEE Proc. on Control Theory & Applications*, Vol. 145, 463 – 471.

Schlenoff, C., Ciocoiu, M., Libes, D., Gruninger, M , "*Process Specification Language: Results of the First Pilot Implementation*," Proceedings of the International Mechanical Engineering Congress and Exposition, Nashville, Tennessee, November, 1999.

Schlenoff, C., Gruninger M., Tissot, F , Valois, J., Lubell, J., Lee, J., "*The Process Specification Language (PSL)· Overview and Version 1.0 Specification*," NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD., 2000.

Schlenoff, C., Gruninger, M., Ciocoiu, M., Lee, J , "*The Essence of the Process Specification Language*," Special Issue on Modeling and Simulation of Manufacturing Systems in the Transactions of the Society for Computer Simulation International, 1999.

Schlenoff, C., Grüninger, M., Menzel, C., Cutting-Decelle, A.F., The process specification language : overview, NWI PSL proposal, 1999

Schlenoff, C., Knutilla, A., Ray, S., "*Requirements for Modeling Manufacturing Process: A New Perspective*," Proceedings of the DETC'97: 1997 ASME Design Engineering Technical Conferences, September 14-17, 1997, Sacramento, California.

Schlenoff, C., Knutilla, A., Ray, S., "*Unified Process Specification Language: Requirements for Modeling Process*," NISTIR 5910, National Institute of Standards and Technology, Gaithersburg, MD (1996).

Scotti R., Lecture Notes for Engineering Management 281 *(Systems Engineering and Management)*, The George Washington University, Washington, DC, 1994.

Seymour, D and Rooke, J., 1995, *The culture of the industry and the culture of research*, Construction Management and Economics 13, pp 511-523

Seymour, D.E , Crook, D & Rooke, J. (1997) The role of theory in construction management: a call for debate. *Construction Management and Economics*, **15**, 117-119.

Shadish, W.R. Cook, T.D. and Houts, A.C. (1986). Quasi-experimentation in a critical multiplist mode. In W. Trochim. (Ed.). Advances in quasi-experimental design and analysis. New Directions in Program Evaluation San Francisco, CA: Jossey-Bass.

Simister, S. J. (1995) Case study methodology for construction management research. Paper delivered at 11th annual ARCOM conference, York, 18th - 20th September

Smith, J., and Heshusius, L. "*Closing Down the Conversation: The End of the Quantitative-qualitative Debate Among Educational Inquirers*," *Educational Researcher* (15:1), 1986, pp. 4-12.

Smith, S.J.J., at al., *Integrating Electrical and Mechanical Design and Process Planning*, Proceedings of the IFIP Knowledge Intensive CAD Workshop, 1996, Carnegie-Mellon University (CMU).

SoftTech Inc., (1981), " *Integrated Computer Aided Manufacturing* (ICAM)" Archıtecture Part II Volume V. Informatıon modelıng manual (EDEF1): DTIC-BO62458.

Strauss A and Corbin J. "*Grounded Theory Methodology - An Overvıew,*" In *Handbook of Qualitative Research,* N. K. Denzın and Y. S. Lıncoln (Eds.), Sage Publıcations, Thousand Oaks, 1994, pp. 273-285.

Suchman, L. *Plans and Sıtuated Actions: The Problem of Human-Machıne Communication.* Cambrıdge, Cambrıdge University Press, 1987.

Susman, G.I. and Evered, R.D "An Assessment of the Scıentıfic Merıts of Actıon Research," *Administrative Scıence Quarterly,* (23) 1978, pp 582-603

Tarandı V, 1998, *Neutral Intelligent CAD Communıcation*—Informatıon exchange ın constructıon based upon a mınımal schema (doctoral thesıs), Royal Instıtute of Technology, Constructıon management and economics, Stockholm.

Tate, A., "Roots of SPAR – *Shared Planning and Activity Representation*", The Knowledge Engineering Revıew, Vol. 13(1), pp 121-128, Specıal Issue on "Puttıng Ontologies to Use" (eds. Uschold, M. and Tate, A.), Cambrıdge Unıversıty Press. http://www.aıaı.ed ac.uk/~arpı/spar/

Tate, A., Planning Inıtiatıve: *Shared Plannıng and Activity Representatıon*: SPAR Homepage, http·//www.aıaı.ed.ac.uk/~arpı/spar/, November 30, 1998.

Taylor, C. "*Hermeneutics and Politics,*" in *Crıtical Socıology, Selected Readings,* P. Connerton (ed), Penguin Books Ltd, Harmondsworth, 1976, pp. 153-193.

T. Froese, M. Fıscher, F. Grobler, J. Rıtzenthaler, K. Yu, S. Sutherland, S. Staub, B. Akıncı, R. Akbas, B. Koo, A. Barron, and J. Kunz. "Industry Foundation Classes For Project Management-A Trıal Implementation", Electronıc Journal of Informatıon Technology ın Construction, Vol.4, 1999, pp.17-36.

Trochım, W. (1982) "*Methodologically based discrepancies ın compensatory education evaluation.*" Evaluatıon Revıew 6, 3: 443-480.

Trochım, W. (1985). Pattern matchıng, valıdıty, and conceptualızatıon ın program evaluatıon. Evaluatıon Review, 9, 5, 575-604.

Trochım, W. (2001). *The Research Methods Knowledge Base.* (2nd Ed.). Cıncınnati: Atomıc Dog Publishing Inc.

Trochim, W. and Linton, R. (1986). Conceptualızatıon for evaluatıon and plannıng. Evaluation and Program Planning, 9, 289-308.

Uschold M. and Gruninger M.. *Ontologies: principles, methods, and applications.* Knowledge Engineerıng Revıew, 11(2), 93--155, 1996.

Uschold, M , *Buılding Ontologies* : towards a unıfied methodology, AIAI, TR 197, 1996

Uschold, M., et al., *"The Enterprise Ontology"*, The Knowledge Engineering Review, 1998, Vol. 13(1), pp. 31-89, Special Issue on "Putting Ontologies to Use" (eds. Uschold, M. and Tate, A.), Cambridge University Press.

van der Aalst, W.M.P., 1999a, *"Formalization and Verification of Event-driven Process Chains,"*Information and Software Technology, 41(10), 639–650.

van der Aalst, W.M.P., 1999b, *"Interorganizational Workflows,"* Systems Analysis, Modeling, Simulation, 34(3), 335–374.

van Leeuwen, J.P. and H. Wagter. 1998. *"A Features framework for architectural information."* In: Proceedings of the Artificial Intelligence in Design '98 conference. pp. 461-480.

Van Nederveen, S and F.P. Tolman, *"Modelling multiple views on buildings"*, Automation in Construction, vol. 1, no. 3, pp. 215-224, 1992.

Vernadat F.B , 1992, 'CIM-OSA - *A European Development for Enterprise Integration* Part2.Enterprise Modelling', Enterprise Integration Modelling: Proceedings of the 1st International Conference, C J. Petrie [ed], The MIT Press, Cambridge, Massachusetts, pp189-204.

Vernadat, F. B. *Enterprise modelling and integration·principles and applications* /F.B. Vernadat Publisher London.Chapman & Hall,1996.

Vidgen, R. & Braa, K. (1997) Balancing interpretation and intervention in information systems research. In: *Information Systems and Qualitative Research.* Chapman & Hall Publications.

W. Reisig, *A Primer on Petri Net Design*, Springer-Verlag, New York, 1992

Walker D H T, 1997, 'Choosing an appropriate research methodology.', *Construction Management and Economics. ISSN 0144-6193*, vol. 15, no.2, March, Bon R, & Hughes W (eds.), E and F N Spon, London, pp. 149-159

Walsham, G. *Interpreting Information Systems in Organizations*, Wiley, Chichester, 1993.

Wisnosky, Dennis E., Allen W. Batteau " *IDEF in principle and practice*," GATEWAY, May/June1990, Pp 8-11

Wix and Liebich 1997 Industry Foundation Classes Architecture and development guidelines. International Alliance for Interoperability (IAI) 1997.

Wix J., 1197. *Standards for Information Exchange and Sharing,* Report prepared for the DoE Construction Directorate Contract Reference ES8/9426, Crown copyright 1997

Wix J., 1998, Industry Foundation Classes: Some Business Questions Examined, ECPPM 1998 Product and Process Modelling in the Building Industry, Watford, United Kingdom.

Wix, J. (1996). *Purpose of a Core Model.* Paper presented to STEP Meeting, Dallas, January 1996

Workflow Management Coalition. Workflow Management Coalition: Reference Model. Document WFMC-TC-1011, Workflow Management Coalition, Brussels, Belgium, 1994.

Wynn, E. *Office conversation as an Information Medium.* Unpublished PhD thesis, University of California, Berkeley, 1979.

Yin, R. K. "The Case Study as a Serious Research Strategy," *Knowledge* (3), 1981, pp. 97-114.

Yin, R. K. (1994). *Case study research : design and methods* (2nd ed.). Thousand Oaks, Calif.: Sage.

Yin, R. K. *Case Study Research, Design and Methods,* 3rd ed. Newbury Park, Sage Publications, 2002.

Yu, K., Froese, T., and Grobler, F (1998), Development of Industry Foundation Classes by International Alliance for Interoperability, Submitted to the proceedings of Computing Congress 98, American Society for Civil Engineering (ASCE), Boston, OCT 1998.

Zuboff, S. *In the Age of the Smart Machine.* New York, Basic Books, 1988.

Zurawski, R. and M. C. Zhou, *"Functional and Behavioral Modeling of Automated Manufacturing Systems,"* in *Handbook of Industrial Electronic Engineering,* CRC Press, 1996, 669-676

Zurawski, R. and M. C. Zhou, "Guest Editorial: Special Section on *Petri Nets in Manufacturing," IEEE Transactions on Industrial Electronics,* 41(6), 565-566, 1994

Software produces an integrated Project Management System specifically designed for the Construction Industry.

List of Publications and other Reports

Conference Papers

Tesfagaber, G., Bouchlaghem, N.M., Baldwin, A.N , Cutting-Decelle, A.F. and Khalfan, M.M.A., "Ontology Modelling of IDEF3 Process Description for AEC Applications Interoperability" , *Proceedings of the Construction Research Congress 2003* , Honolulu, Hawaii, March 2003 , [CD-ROM].

Cutting-Decelle, A.F., Pouchard, L., Tesfagaber, G , "*Ontologies and Standards-Based Approaches to Interoperability for Concurrent Engineering,* " In Concurrent Engineering Construction Projects, Anumba, C.J., Kamara, J.M., Cutting-Decelle, A.F. Eds., 2004.

Tesfagaber, G., Anumba, C.J., Bouchlaghem, N M., Cutting-Decelle, A.F. and Baldwin, A.N., "Semantic Process Model for Applications Integration in Architecture, Engineering and Construction" , *Proceedings of the International Workshop on Information Technology in Civil Engineering* , International Workshop on Information Technology in Civil Engineering , Washington, USA, November 2002, pp 318-335 .

Cutting-Decelle, A.F , Tesfagaber, G., Anumba, C.J., Baldwin, A.N. and Bouchlaghem, N.M , "PSL A new language for exchanging process information throughout the construction process" , *ECPPM02 International Conference* , ECOOM02 International Conference , Portoroz, Slovenia, September 2002, pp 215-222 .

Cutting-Decelle, A.F., Anumba, C.J., Baldwin, A.N., Bouchlaghem, N.M. and Tesfagaber, G , "PSL A common language for the representation and exchange of process information in construction" , *Perspectives on Innovation in Architecture, Engineering and Construction* , Anumba, C.J, Egbu, C, and Thorpe, A., Centre for Innovative Construction Engineering (CICE) , Loughborough University, Leics , UK, July 2001, pp. 865-878 .

Unpublished Journal Paper

Tesfagaber, G., Bouchlaghem, N.M., Anumba, C.J., Baldwin, A.N., Cutting-Decelle, A.F , "Automation of Process Information Exchange between AEC/FM software Applications"

Reports on Related Projects

Cutting-Decelle, A.N., Bouchlaghem, N.M., Tesfagaber, G., Anumba, C.J , A F., Baldwin, LEXIC Project Towards a Common Language for the Representation and EXchange of Process Information; **DELIVERABLE 1:** A contribution to the Integration of Concurrent Engineering in Construction, May 2002, EPSRC Grant GR/N22526.

Cutting-Decelle, A.N., Bouchlaghem, N M., Tesfagaber, G., Anumba, C J , A.F., Baldwin, LEXIC Project Towards a Common Language for the Representation and EXchange of Process Information; **DELIVERABLE 4:** REQUIREMENTS FOR AN INTEGRATED AND GENERIC PRODUCT AND PROCESS MODEL FOR CONSTRUCTION March 2002, EPSRC Grant GR/N22526

Cutting-Decelle, A.N., Bouchlaghem, N.M., Tesfagaber, G., Anumba, C.J., A.F., Baldwin, LEXIC Project Towards a Common Language for the Representation and

EXchange of Process Information; **DELIVERABLE 5** :CONCEPTUAL MODEL OF THE SOFTWARE PROTOTYPE, June 2002, EPSRC Grant GR/N22526