

A data mining integrated architecture for shop floor control

Srinivas and J A Harding*

Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, UK

The manuscript was received on 13 August 2007 and was accepted after revision for publication on 22 January 2008.

DOI: 10.1243/09544054JEM980

Abstract: The dynamics of the market are forcing shop floor control systems to enhance their intelligence and reactivity to change. The control system must also have mechanisms to process the vast amount of data, which is generated with the application of cheap data logging systems and should also incorporate a mechanism for continuous learning. Data mining provides a mechanism for analysing data and generating new knowledge. This paper presents an intelligent knowledge-based decision support system for shop floor control, which incorporates data mining techniques and intelligent agent technology to provide useful information, knowledge, and a mechanism for continuous learning. The functioning of the system is described and the architecture of different agents that constitute the system has been discussed in terms of the different modules required. A process to integrate the individual data mining agents associated with the different processes has also been described. Finally a prototype of the system is presented to demonstrate the application of this agent-based system.

Keywords: shop floor control, data mining, decision support system, agent technology, continuous learning

1 INTRODUCTION

The shop floor represents a fundamental part of a manufacturing enterprise, where value adding processes take place, transforming raw materials into final products. Shop floor control (SFC) manages the operations responsible for transforming planned orders into a set of outputs through activities such as reactive scheduling, order release, resource allocation, online process planning, data collection, monitoring, and so on. Shop floor manufacturing environments consist of a number of interconnected systems with operations that are interdependent from a decision-making perspective. The large number of components found on the shop floor require protocols for interaction and integration of control components and reference architectures to specify their dependencies, interfaces and interaction mechanisms, constraints, and rules for different activities. SFC plays an important role in coordinating the operations necessary to process production orders across manufacturing resources [1].

The main objective of this control is effective and efficient usage of resources [2]. A literature review of SFC revealed the ineffectiveness of deterministic and long-range solutions in real-time manufacturing problems involving various uncertainties such as machine failures, breakdown, and so on. However, assessment of past and current research efforts in manufacturing control reveals that the key assumptions of any particular control paradigm cause the inherent drawbacks on the general applicability of that paradigm. The complex challenges posed by uncertainties on the shop floor have motivated researchers to look at intelligent and adaptive techniques to carry out real-time manufacturing.

The shop floor is an important unit of any manufacturing system and has offered challenging problems for researchers and practitioners in the last few decades [3, 4]. Most research has been aimed towards the development of mathematical models [5–8], heuristics [2, 9, 10], and knowledge-based systems [11–13], each of which has met with varying success. Mathematical models can provide optimal solutions, but cannot be implemented on large-scale problems because of their inability to provide solutions in a reasonable amount of time [14]. Heuristic approaches can deliver very fast solutions but tend to be myopic in nature [15]. Some dispatching rules

*Corresponding author: Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Ashby Road, Loughborough, Leicestershire LE11 3TU, UK. email: j.a.harding@lboro.ac.uk

such as the shortest processing time and earliest due date, are popular and provide good results [9, 16]. However, these rules independently cannot consider real-time information and are unable to consider parallel and alternative process plans. Knowledge-based systems have been successful to some extent as they can help shop floor managers to consider various alternatives quickly. The major disadvantage of knowledge-based systems is that they do not scale up well because they become disordered and search intensive as they increase in size. They also have poor conflict resolution capabilities and inherit some of the problems of heuristic approaches. Therefore an intelligent shop floor controller for a dynamic environment must (according to [4, 8, 13]):

- (a) incorporate mechanisms, which monitor the environment in real time;
- (b) present relevant information to managers so that they can make timely informed decisions;
- (c) support learning mechanisms that are robust in the face of various changes in the environment.

These requirements provide a context for the current research, as they indicate that better knowledge discovery and knowledge management are required to improve SFC. Data mining tools and techniques have been explored during this research as they provide an methodology for analysing real-time data and can generate useful information and knowledge in many areas important for SFC.

To address the requirements for modern SFC systems an intelligent decision support tool is proposed, which incorporates data mining and intelligent agent technology to provide useful information and knowledge for an SFC system. The architecture presented is similar to the architecture presented by Wang [17] that takes advantage of the intelligent, autonomous, and active aspects of agent technology. It also provides additional functionality through the integration of data mining processes for the generation of required knowledge and information for different activities on the shop floor into a decision support framework by applying intelligent agent technology.

Enterprises often rely heavily on the expertise and experience of their employees in decision-making processes, yet do not fully exploit other less clearly defined experiences and information that may also exist in their recorded product life cycle data. The relationships between operational variables and performance are usually not thoroughly analysed and interpreted by enterprises. Effective knowledge discovery approaches could enable organizations to identify, maintain, and exploit their dispersed knowledge sources and experience for the long-term benefit of the corporation, through improved shop floor systems [18]. The two prime sources of manufacturing knowledge in an organization are its employees

and its operational data, which are supported by secondary sources such as technical specifications, plant practices, and operating theory. The capture of human knowledge is generally achieved through knowledge-elicitation using many established techniques. However, useful knowledge may not all be well-known by human experts, or be immediately obvious or visible: it may lie 'hidden' as patterns within the plant data. Data mining techniques have been recently used to extract such knowledge from plant data and have consequently helped in validating and providing alternative sources for experts' knowledge [19–21].

Knowledge is the most valuable asset of a manufacturing enterprise, as it enables the business to differentiate itself from competitors and compete efficiently and effectively to the best of its ability. In the competitive environment, organizational knowledge is not perpetual, but has a life cycle. The value of organizational knowledge deteriorates with time owing to changes in the competitive environment. Therefore, the major activity of manufacturing firms is no longer confined to production but lies in the systematic management of knowledge rapidly to meet customer demand [22]. Another key reason for deployment of knowledge management systems is the recent advances in information technologies that enable firms to build systems that integrate and consolidate experts' experiences, thereby enabling the companies to provide better services to their customers. Therefore, knowledge management becomes a crucial tool for corporations to survive in the volatile marketplace and to achieve a competitive edge [23]. Development of a knowledge-based system to support the decision-making process is also justified by the inability of decision makers to diagnose efficiently many of the malfunctions that arise at machine, cell, and entire system levels during manufacturing [18].

Manufacturing enterprises routinely generate large amounts of data during their normal operation and the current research is built on the belief that these data stores can be valuable assets and potentially important sources to explore for new information and knowledge. By exploiting these assets and becoming more data aware, manufacturing systems can respond more quickly to market changes and challenges in their business and production activities. This may be achieved by intensive and intelligent analysis of existing databases with the objectives of identifying new trends, to predict results and improve performance. Hence the proposed SFC system should be able to apply data mining technology to extract and exploit valuable knowledge from its existing and historical operational databases.

Although potentially very valuable for improved exploitation of knowledge assets, knowledge discovery and data mining can be complex, time-consuming, and expensive processes. A review of manufacturing

applications of data mining [24] showed that these are mostly 'one-off' applications or experiments and no research has yet been reported into a generic data mining enabled architecture that would be relevant across different domains. This research therefore attempts to fill this research gap by proposing a new data-mining-enabled architecture for manufacturing – SFC. A data mining agent (DMA) has been integrated into a multi-agent architecture of SFC. The purpose of the DMA is to extract useful knowledge from large datasets obtained from product life cycle data and store them in a knowledge pool which can be further reused. Knowledge generated from mining enterprise-wide data can result in a better understanding of the consequences of decisions made at all levels of the company. The combination of a reusable knowledge pool and the automatic discovery and update of knowledge that is generated through data mining analysis should provide managers with the information that they require for decision making on the shop floor. The classification model based on data mining algorithms [25–27] can support various functionalities of SFC. It should also be noted that this research does not provide a detailed specification of the hardware or software that may be required for the implementation of such a system, but it provides a conceptual description and explanation of a set of modules that can be combined to develop a data-mining-enabled system for decision support in SFC.

The rest of this paper is organized as follows. The next section presents the literature survey in the area of SFC, which highlights the shortcomings of many existing solutions and in data mining, which shows the under-exploited opportunities offered by these approaches. Section 3 presents the proposed architecture and this is then detailed further by a prototype implementation which is discussed in section 4. The last section presents the conclusions of this work.

2 LITERATURE SURVEY

The decision support system for SFC that is proposed in this research integrates the application of data mining for ongoing learning and improvement of knowledge within decision support for SFC. Hence to provide a context and background for this work, the next two sub-sections present a literature survey of SFC and data mining.

2.1 Shop floor control

SFC is a complex process which lies at the heart of operations for manufacturing companies, and primarily involves job scheduling, progress monitoring, status reporting, and corrective actions [28, 29].

SFC has rapidly to reflect the current system status to allow job processing to be controlled in a real-time mode. However, the manufacturing system behaviour, which is an accumulation of status in time, is highly dependent on the control system architecture, control function classification, and allocation of controls to different control levels (i.e. control allocation). All this has had a great effect on the way production facilities are designed and managed with large changes in manufacturing systems, in levels of automation, and in the application of computer-based technology at all levels of the company.

Existing modelling frameworks for manufacturing system control can be classified into hierarchical, heterarchical, and recent control frameworks. Diltz *et al.* [30] give a detailed overview of the evolution from centralized control, over hierarchical and modified hierarchical, to heterarchical control. Hierarchical control architectures have a strong association with the software structures of traditional control architectures and are characterized by a fixed structure and large volume of global information [31]. The motivation for such architectures is derived from the natural hierarchy existing in all complex organizational systems. In these architectures, a master module co-ordinates the operations of all the facilities. The communications between the master and its subordinate modules are carried out by passing commands and receiving feedback from the subordinate modules through well-organized and defined communication protocols. An advantage of hierarchical control is that it allows the control problem to be divided to limit the complexity of the entire structure [32]. These implicit assumptions are the cause of the main drawbacks of hierarchical architectures such as long lead times, high inventory, tardiness, and handling of runtime errors [33].

Hatvany [34] suggested the need for a manufacturing control model, which allows total system analysis from incomplete knowledge, provides automatic recognition and diagnosis of fault situations. Therefore, heterarchical architectures have been suggested, since these reject a master/slave relationship and provide full autonomy, with the autonomous components cooperating in order to reach global objectives through local decision making. These autonomous components are often referred to as agents. Nwana *et al.* [35] described four co-ordination techniques: organizational structuring, contracting, planning, and negotiation. MacChiaroli and Riemma [36] extended the negotiation process, which is iterative in nature and forces a convergence between demands and offers. Dewan and Joshi [37] and Veermani *et al.* [38] presented an auction-based mechanism that can be used for scheduling within a distributed decision-making environment. However, the basic assumption of independence of agents, which prohibits the use of global

information can be a drawback and obstructs their widespread application in some industrial application areas. The independence of agents makes central scheduling and resource planning a difficult task. The control system cannot guarantee a minimum performance level for cases outside the scope for which the rules are tuned.

Hierarchical and heterarchical control systems have some desirable characteristics but also have limitations. Several attempts have been made to capture the positive aspects of both with varying degrees of success. These have mostly been focused on the simple goal of enabling the manufacturing system efficiently to survive and adapt to the ever-changing manufacturing environment. Maturana *et al.* [39] proposed MetaMorph, a multi-agent architecture for a distributed manufacturing system. They suggested two types of agent: a resource agent for physical resources and a mediator agent for co-ordination. McFarlane and Bussmann [40] review the use of a holonic manufacturing control system in production planning and control. Shen *et al.* [41] presented an intelligent SFC system based on the internet, web, and agent technology. Lima *et al.* [2] proposed an agent-based production planning and control system that can be dynamically adaptable to local and distributed utilization of production resources and materials. Shin and Cho [8] proposed a rapid development methodology that fulfils given characteristics of a shop floor through a formal model-based control software specification. Lea *et al.* [42] presented a multi-agent enterprise resource planning (ERP) system that utilized software agents to achieve system-wide integration.

Performance of the manufacturing system relies heavily on the control strategies adopted. The modelling of manufacturing control can be classified into three different groups: mathematical, heuristic, and knowledge-based approaches. In the mathematical approach, multi-dimensional spaces are searched for the 'best' solutions to provide optimal results for given decision criteria. The results are dependent on the weights given to each individual decision criteria as the assigning of weightings, such as cost or set-up time is often arbitrary and relies on value judgements, leading to results that are often less than optimal. Another problem with this approach to scheduling is that the computational efforts can be vast owing to combinatorial explosion. Duffie and Prabhu [33] exploited constraints to generate a solution. Bongaerts *et al.* [43] used partial derivatives for modelling SFC. Wullink *et al.* [44] presented a mixed integer programming model for SFC. Simulation has also been widely used for decision making in SFC (Roy *et al.* [45] and Lastra and Colombo [46]).

Most manufacturing systems have large numbers of machines and many jobs with various routings

competing for the various resources and so an algorithmic solution to the manufacturing control problem is not possible. In these instances heuristics or 'rules of thumb' are often used, and these evolve over time through trial and error and are based on past experience of successful decisions. Lin and Solberg [47, 48], Tilley [49], MacChiaroli and Riemma [36], and Shen *et al.* [41] used market-based pricing heuristics for modelling SFC. Lima *et al.* [2] applied a contract-net heuristic methodology for SFC modelling. However, these methods cannot independently consider real-time information and are unable to consider parallel and alternative process plans.

The use of expert systems within control can be seen as an extension of the use of heuristics, where the selection of the rules to apply are suggested by the knowledge-based system working on the encoding of an expert's domain specific knowledge. Metaxiotis *et al.* [50] presented a survey of expert systems and Iwamura *et al.* [51], Sun *et al.* [52], Shiue and Guh [13], Shen and Norrie [53], and Ozbayrak and Bell [18] all used knowledge-based approaches for SFC modelling. Pham and Afify [54] reviewed the application of machine learning techniques in manufacturing. These systems have been successful to a certain extent, as they help shop floor managers to consider various alternatives quickly. However, the major disadvantage of knowledge-based systems is that they do not scale up well as they become disordered. They have poor conflict resolution capabilities and inherit the disadvantages of heuristic approaches and updating the knowledge base is a tedious task.

Analysis of existing reference architectures has revealed some important shortcomings and to avoid these the control architecture should address the issues of dynamic structure, decoupling of structural aspects from the control algorithm, reactive scheduling and process planning, generic applicability, adaptability, and efficient knowledge management and discovery. The main area addressed in this work has been to provide a methodology for knowledge discovery within the SFC system from data recorded during different operations in the product life cycle.

2.2 Knowledge discovery and data mining in manufacturing

The key asset of any manufacturing enterprise is its knowledge, and to exploit this resource properly it needs to be maintained, revised, and at times replaced by new knowledge. Modern manufacturing businesses store most of their data and knowledge electronically (although some tacit knowledge will only exist within their employees). Thus, routinely collected electronic data can be a source of valuable knowledge that is implicitly coded within them. In manufacturing, these data might capture performance and optimization

opportunities, as well as the keys to improving processes. Data mining research in manufacturing contexts has primarily focused on attempts to identify, extract, and make explicit knowledge that may lie hidden within electronic files or databases, by using various statistical or artificial intelligence techniques and algorithms.

The use of databases and statistical techniques is well established in engineering [55]. The first applications of artificial intelligence in engineering in general and in manufacturing in particular were developed in the 1980s [56, 57]. The scope of these activities, however, has recently changed. Current technological progress in information technology (IT), data acquisition systems, and storage technology enables the storage and access of large amounts of data at virtually no cost. However, the main problem in an information-centric world remains how properly to put the collected raw data to effective use.

Harding *et al.*'s [24] review of approximately 90 papers relating to research into data mining applications in manufacturing shows that data mining methods have been successfully introduced in many fields. Data mining in manufacturing is still a research topic, but industry is also increasingly showing interest in these techniques in order to solve their real-world problems.

Many reported applications are related to the identification of causes of malfunctioning of different types of manufacturing system or process, and hence the discovered knowledge should lead towards the better functioning of the manufacturing enterprise. Most reported applications in data mining have been 'one-off' single shot experiments. While these are useful for solving particular problems, the knowledge obtained in these projects may not be fully exploited unless the lessons learned in one project are also utilized in other projects and the knowledge generated should also be integrated in the manufacturing system. Future work in this area will be directed towards integration of data mining approaches at different levels in the manufacturing enterprise and utilizing the results obtained at those levels. These approaches should be able to use diverse data located at different places and integrated in the database management system. Future work should also be directed more towards optimizing the process and suggesting pre-emptive measures rather than just predictions.

3 PROPOSED ARCHITECTURE

The application of multi-agent systems (MAS) based on the concept of distributed artificial intelligence is considered to be the most promising architecture for next-generation manufacturing [58]. These consist of

distributed heterogeneous agents and make use of flexible control mechanisms for creating and co-ordinating the resulting society of agents. This society of agents provides the foundation for the creation of an architecture that possesses the capability to benefit manufacturing by enhancing a system's reliability, maintainability, flexibility, fault recovery, and stability, as well as providing a means for real-time decision making on the shop floor [59].

Manufacturing control involves the co-ordination of the flow of both physical items and information. Therefore, the agents within an MAS in this context are found to represent either the physical or informational entities that are required by the system. Agents must be able to participate in a society governed by some set of basic protocols if they are to perform the necessary tasks to achieve their objectives. To enhance the decision-making process, the proposed system makes use of quasi-heterarchical structures, which essentially add one or more layers to the single-layer heterarchical architectures. These additional layers are for the purpose of creating agents with more global awareness, allowing them to take the role of manager or mediator of subsets of the agent society [59, 60].

Decision support systems are computer-mediated tools that assist decision making by presenting information and interpretations for various alternatives, enabling decision makers to make more effective and efficient decisions. Incorporating data mining techniques in a decision support system can aid the decision-making process through a set of recommendations reflecting domain expertise [61]. It provides information and knowledge which can be crucial for the decision-making process. It has been shown that the best way to capture and share knowledge is to embed it in the jobs of workers and to ensure that learning new knowledge is not a separate task that requires additional time and effort to encode what can or has been learnt from others [22]. In order to build a knowledge repository that captures and embeds the value-added knowledge and enhances decision making, it is essential to build a knowledge-based system with data analysis capability. It is difficult to capture and embed informal knowledge that resides within minds of people in the system. McDonnell Douglas, which is now part of Boeing, tried to develop an expert system that contained the expert knowledge necessary to determine whether an aircraft is positioned properly for landing. They gathered the human knowledge by interview and observation. The system took two years and a tremendous amount of resources to capture the human expertise and demonstrated how difficult it is to capture and embed tacit knowledge in a system [62]. However, it should be less time consuming and require less resources to develop a system that

captures and embeds structured knowledge [22]. In the proposed system, product life cycle data (product, process, and machine data) can be used to generate structured knowledge, with the primary aim that the results generated by the decision support system should be at least as good as the decisions made by domain experts. The proposed DMA analyses data and generates knowledge by capturing knowledge embedded in the existing databases.

The multi-agent based SFC system that has been designed in this research has a decentralized control mechanism which uses knowledge, information, and alternatives provided by the decision support tool. The knowledge pool can be generated and updated by the DMA, which reduces complexities, enhances flexibilities of functioning, and addresses the problem of dynamic management in real time by exploiting the various flexibilities offered by this system and provides various possible alternative measures to be taken in different scenarios. The proposed system utilizes the performance information provided by the knowledge pool and the DMA to select the optimum alternatives. Figure 1 shows the proposed architecture for SFC consisting of multiple autonomous agents that are interconnected with the shop floor controller via a communication bus.

The functional characteristics of the constituent agents in the proposed control architecture are outlined below.

1. *Process planner agent.* The process planner agent dissects the customer orders into a set of alternative manufacturing operations and these alternative operations are represented with the help of a logical AND/OR graph. The proposed approach aims to minimize early commitment by constructing a process plan based on the most recent state of the shop and by using knowledge representations which are open for alterations.
2. *Scheduling agent.* In this context scheduling explicitly refers to the optimization process of looking for the best schedule. It traditionally generates a Gantt-chart, which represents the planned operations on every machine/workstation. This information may be used by the other agents to improve their logistical decision-making process. The scheduling agent performs the above task and sets guidelines for the other agents to follow as exactly as possible. The scheduling agent may have many resource allocation algorithms, each generating a schedule, but it selects the 'best' depending on the goal of the organization.

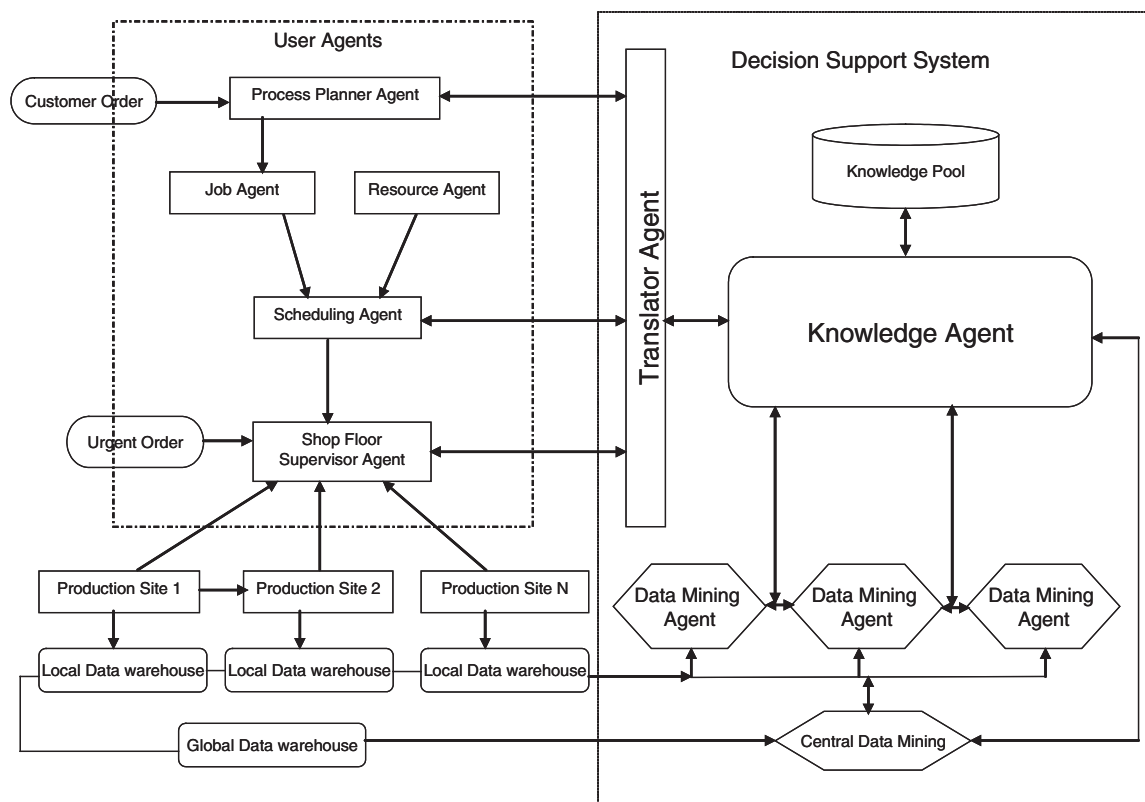


Fig. 1 Proposed agent-based system

3. *Shop floor supervisor agent.* The basic function of the shop floor agent is to follow the schedule generated by the scheduling agent as closely as possible and to determine feasible actions in case of breakdowns and urgent orders. The local schedules are compared pair wise with a check for potential conflicts, which are resolved using the rules generated by an operational data warehouse.
4. *Resource agents.* These represent the basic components of the manufacturing system, and bridge between the abstract world (control world) and real world (physical world). The degree of complexity depends on the physical resources represented. Information is given about the state of any of the associated resources and estimates of the time during which the resource will remain in a given state.
5. *Job agents.* These are created by the process planning agent according to the production objectives and have information about the resources they will use and also the sequence of operations they will perform. These agents also have information about the standard times for manufacturing operations, and thus, they can send an alarm message if these times are exceeded.
6. *Data mining agents.* These are responsible for accessing data and extracting higher level useful information from the data. These agents will perform the various steps of the data mining process and will then select suitable algorithms from the library to provide the required information online and generate knowledge for updating the knowledge pool. The information and rules that they generate will therefore be utilized, shared, and reused by other agents in decision making.
7. *Translator agent.* This acts as the intelligent interface agent between the decision makers (process planning agent, scheduling agent, supervisor agent) and the decision support system. It provides the interaction between any particular decision maker and the knowledge agent.
8. *Knowledge agent.* The knowledge agent provides system co-ordination, facilitates knowledge communication, and evaluates the results obtained by the DMA before updating the knowledge pool. When a decision maker requests support through the translator agent, the knowledge agent interrogates and evaluates the available knowledge in the knowledge pool to identify what relevant information and rules may be known about the current problem and what advice it might provide to help prioritize options for selection. It then provides the required knowledge and information back to the decision makers via the translator agent.
9. *Knowledge pool.* This consists of domain knowledge rules and expertise generated through data

mining but verified and evaluated by human experts and then structured in ways that can be utilized by the decision support system [63].

The proposed system organizes the agents' functionality into three layers (Fig. 2). Like most multilayer architectures, as one moves up the architectures the planning horizon is extended and the decisions are more global in scope. The top two layers of the architecture support the deliberative behaviour of the agents with the bottom layer providing their reactive behaviour.

The global planning layer (process planning agent) is responsible for performing the higher level planning functions necessary to ensure that the agent is contributing to the goals of the overall manufacturing system. This job involves formulating local objectives that account for the needs and current actions of the environment within which the agent operates. The performance of this job requires that the global planning layer maintains an awareness of the overall manufacturing system through the use of a system model and communicates with system agents. The system model is used to estimate the effect of the impact of local decisions on the overall system. The global layer within an agent is also involved in initiating and managing interactions with other agents to effect global changes and satisfy system objectives.

The local objectives created by the global layer are passed down to the second layer (scheduling agent), which is the local planning layer, where the planning functions determine the actions necessary to achieve these objectives. These two layers have the information about the shop floor environment and create plans of actions to meet its demands, thus providing the deliberate behaviour for this agent system. These actions are passed to the behavioural layer (shop floor supervisor agent) in the form of commands that the agent will carry out to effect the necessary system changes. In addition to executing the required actions, the behavioural layer is also programmed to monitor its environment and respond to events directly affecting it. The behavioural layer would employ schemas to provide the capability to connect perceptual signals with actions providing for the reactive behaviour of this agent system.

In the proposed model, the process planning agent, job agent, resource agent, scheduling agent, and shop floor supervisor agent have all been considered as user agents. DMA is a part of the decision support system which provides information, knowledge, and alternatives to the queries provided by these agents. The decision support system communicates with these agents organized in different layers by providing the different information required by its domain. The DMA for each layer uses different data sources to extract the required information and provide a set of

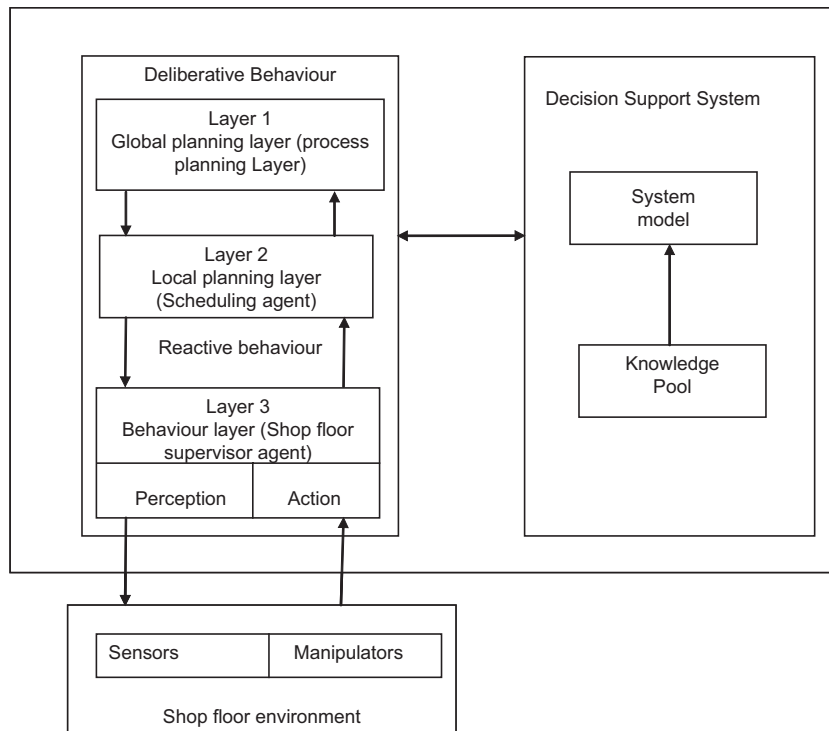


Fig. 2 Hierarchy of shop floor control environment (adapted from Bauer *et al.* [28])

recommendations reflecting domain expertise. In this way it provides useful features for the application of domain knowledge in decision making and should be able to provide information for re-routing, process parameter variables, and feedback for statistical process control, etc., to the decision makers. The decision support system should also be able to provide performance information generated in those methods to assist decision makers in the system when there are alternative possible choices. The knowledge agent provides the performance information with different alternatives either by sifting through the knowledge pool or by invoking the DMA. The application of data mining can potentially help in generating knowledge autonomously and the knowledge generated in one data mining application can also potentially be reused in future.

The main contribution of this research is the incorporation of data mining techniques within a decision support system in SFC. Therefore the DMA, translator agent, knowledge agent, and knowledge pool agent are discussed further in later sections. The descriptions of other agents are widely available in literature and hence have not been pursued further in this paper.

3.1 Components of a data mining agent

Data mining is a process of extracting useful knowledge from data automatically. It combines the tools and techniques from machine learning, statistics, artificial intelligence, and data management. It tends

to simulate human know-how and intelligence into a knowledge base, which can then be used to reason through and determine possible solutions to a problem. The primary purpose of the proposed DMA is to extract as much useful knowledge as possible from existing operational data to build, populate, and update the knowledge pool so that it consists of knowledge and experience from the manufacturing environment that can be regularly and repeatedly shared and reused to assist ongoing decision making. At present, successful data mining tends to rely on a combination of experience and expertise from both data mining practitioners and domain experts (i.e. it is only a semi-automatic process, relying on some human interaction). However, it can still be used to generate useful information about the system performance based on current status data and past records.

A DMA must be able to perform the following functions in order to achieve its main objectives in knowledge discovery. It must be able to:

- collect appropriate data;
- prepare the data so that they are free from errors and structured appropriately for the application of particular data mining techniques;
- determine and apply prior knowledge (if available) to simplify, facilitate, or accelerate the main data mining tasks;
- apply an appropriate range of data mining algorithms;

- (e) analyse results and determine next appropriate course of action (which might be to stop, if suitable results have already been obtained, or if not, to continue with further data mining).

These steps may be iterative, i.e. the previous functions may be invoked again depending on the requirements of the problem and algorithms selected for data mining.

To achieve the above functionalities, it is proposed that the DMA should consist of the following modules (see Fig. 3).

1. *Data collection module.* There are many different automated methodologies that companies can use to store their product life cycle data in files and databases. In the proposed system, there is a local data warehouse for each station and this is linked to the global data warehouse. The data collection module performs the following two tasks:
 - (a) *Define data needs.* The data required for mining depend on the type of information sought. The relevance of attributes will differ from problem to problem, so, for example the measurements of a component might be indispensable information in solving one data mining problem (e.g. to provide knowledge of accuracy achievable from particular manufacturing resources and set-up), but might not be essential for a different problem (such as to provide knowledge of typical availability of a particular type of resource).
 - (b) *Data acquisition.* The initial step in any data mining approach is the selection of a historical dataset for analysis. There are three

main considerations within this step: data accessibility, population of required data attributes, and data heterogeneity. The accessibility of data can be revoked for several reasons, e.g. data might not be stored electronically or may not be physically accessible. Population of relevant attributes is crucial to the quality of discovered knowledge. Null values can have some semantic meaning, but they degrade the quality of result. Data heterogeneity is a major concern when there are different data sources. Semantic inconsistencies have to be considered in case of data heterogeneity. Data-exporting methods therefore have to accommodate these inconsistencies. Domain expert knowledge needs to be employed in this step.

2. *Data cleaning and pre-processing module.* Data cleaning is a time-consuming but essential operation in any data mining task. The process of data cleaning depends upon the particular data mining task and objective. The raw data collected from actual manufacturing processes may contain impurities, missing values, and outliers. The data cleaning procedure attempts to remove noise, outliers, and duplicate records. It takes values from bills of materials and process specifications to determine the specified or required operating value (or range of values) and values lying outside these ranges are treated as noise for process variables. Any irrelevant and redundant values for the process variables should be removed as they can potentially deteriorate the modelling results. The steps required to 'clean'

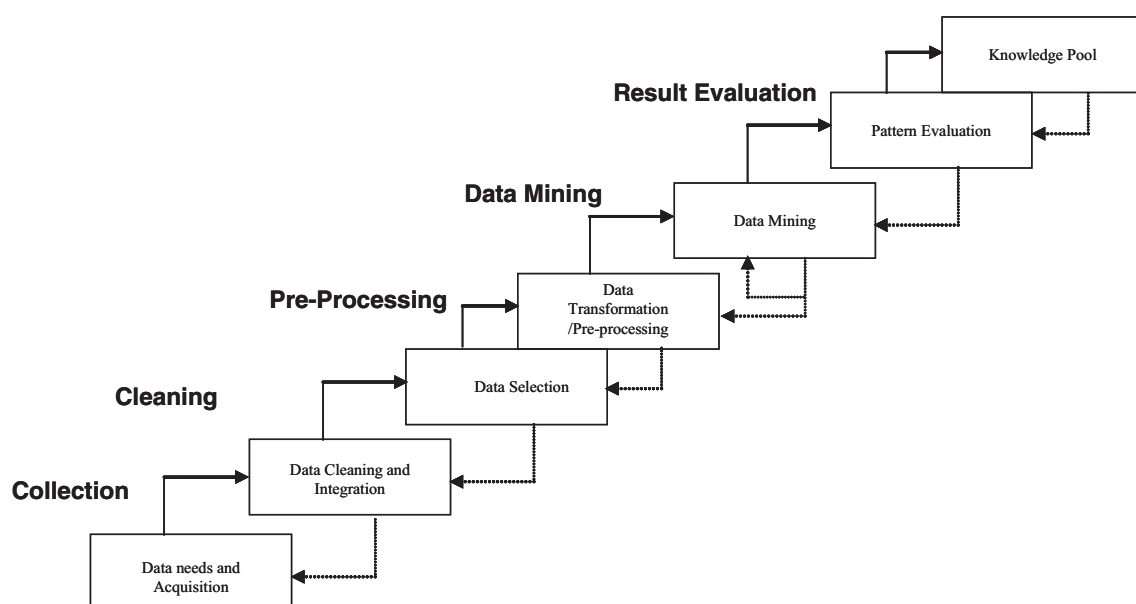


Fig. 3 Main steps in knowledge discovery

the data will vary with problem domain and knowledge sought. After the initial cleaning, the data will need to be transformed into different formats depending upon the algorithm selected for data mining. Data pre-processing is a significant and important task in data mining as it may affect the data mining algorithm's efficiency and accuracy. Data pre-processing may consist of data clustering, predicting and filling in missing values, coding and heterogeneity resolution, etc. Data pre-processing also includes any steps needed to transform data into a different format if this is required by the chosen algorithm. Data cleaning and pre-processing is an iterative task. Data cleaning and transformation is also carried out after identifying critical variables and algorithm selection. The common data transformation tasks are data smoothing, normalization, categorization, etc. These tasks may also be carried out after the selection of particular data mining algorithms as individual algorithms may be dependent on different forms of data transformation.

3. *Determination of critical variables module.* Dealing with a large number of process variables has several disadvantages. First, optimizing all the variables takes too much computing time. Second, adjusting all the optimized process variable at the manufacturing facility takes too long. One type of information, which is usually irrelevant, is primary keys, since they are unique by nature and thus do not contain any patterns. To determine the critical variables, the process variables can be ranked according to their importance index values using different multivariate statistical techniques, principal component analysis, OLAP, or discriminant analysis [64]. In addition to the mathematical indexes the domain knowledge is also employed. This domain knowledge can be embedded with the help of humans or can be based on the knowledge gained in similar data mining tasks. The identification of critical process variables makes it possible to achieve the desired levels of quality by optimizing only a small number of controllable process variables.
4. *Mining module.* Data mining algorithms are then applied on the dataset to characterize and identify the different interrelationships among variables. The rules thus generated need to be checked for their validity and quality. Comprehensive information can then be released to assist the decision makers. The mining algorithm predicts the various outcomes of the decision and also estimates the limitations of different parameters that existed in the decision. The mining module selects the algorithm to be applied on the dataset depending on the data

type, information required, and other constraints. The rules thus obtained are statistically validated and after consultation with the subject matter expert (human expert) they are added to the knowledge pool. The human expert also needs to analyse the rules generated and validate them before storing them in the knowledge pool.

5. *Result evaluation module.* The high-quality model (results) generated from the above module are evaluated before being stored in the knowledge pool for future use. In this step, various assumptions are reviewed to check if the business objectives have been sufficiently considered while performing the different data mining steps.

3.2 Data mining agent

Manufacturing databases are dynamic as new records are added regularly. The data thus obtained must be monitored without interruption so that faults can be diagnosed and eradicated immediately. Therefore, the DMA (knowledge discovery process) must run concurrently with the production process [65]. The DMA must also provide multi-dimensional views on scattered manufacturing data which might exist in several different types of file or database and generate aggregated data from these to gather and provide the information for further assessment to whichever agent has requested information and support within the SFC system. The DMA therefore potentially needs to analyse data from any of the system repositories which store product life cycle data.

The operational facility and its data interface are shown in Fig. 4. The knowledge agent interface manages the communications between the DMA and the knowledge agent. A communication is a message based on a shared understanding of the domain called an ontology. Ontologies are conceptualizations of a domain in a form which can be understood by all the agents involved and by careful design they can remove ambiguity from the communication language. The detail and design of such ontologies are beyond the scope of this research; however, details can be found in Lin *et al.* [66]. The interface agent translates the messages received from the knowledge agent from the common format into the local format based on the ontology and also converts the messages from the DMA into the common format before transmitting them to the knowledge agent. The required knowledge to perform the mining task, common vocabulary, information about the different users, and domain knowledge are stored in the agent knowledge base. This knowledge drives the different functionalities of the DMA while the data interface provides a means for communication with different databases, possibly accessing them through the

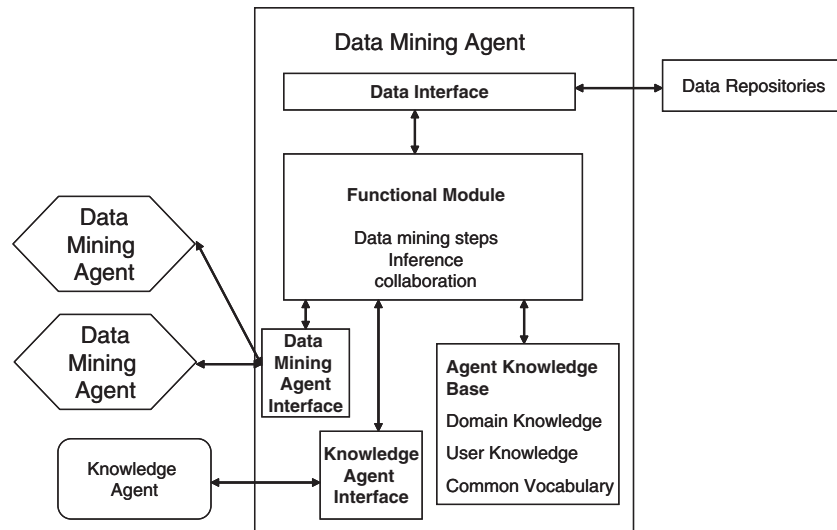


Fig. 4 Data mining agent architecture

structured query language. The DMA interface allows the DMA to collaborate with other DMAs. The knowledge generated about the data mining process/stages (e.g. strategies for data cleaning, transformation, etc. can therefore be communicated to other DMAs and this helps in reuse of knowledge generated in applying data mining tools. The functional module provides the basic functionality of this agent. It carries out the task of data analysis and knowledge generation. This module consists of various sub-functions that are required to carry out the data mining task such as data cleaning, data transformation, result evaluation, etc. The tasks performed by this module depend on the information received from the knowledge agent interface and it uses the knowledge stored in the agent knowledge base module for its functionality. After completing the task of data mining, the functional module communicates its results through the knowledge interface and the functional module is then terminated. The DMA needs to be able to utilize data mining tools, techniques, and methodologies to perform a wide range of data mining applications.

3.3 Integration of data mining

The myriad of organizational informational sources require a team of data miners distributively to discover data relationships or patterns that are contained in the information sources. The individual data miners will need to exchange the captured knowledge between themselves and make such discovered information available to decision makers (via the knowledge pool). The DMAs therefore need to be able to operate in at least two contexts: first to provide the results that are of importance to any particular decision issue at hand and provide assistance for an

individual instance of the organizational decision-making process; and second to capture data relationships with respect to some general knowledge objectives, to regularly maintain, and update the knowledge pool with accurate and valid knowledge, which requires that the data mining processes are executed regularly after certain intervals.

The DMA can be applied at different micro levels, i.e. within different sections of the manufacturing operation and also at a macro level, i.e. to examine the overall manufacturing processes. A central DMA can also be created which co-ordinates, shares, and exchanges data and knowledge with other similar DMAs. Thus a network of DMAs can be built by connecting different DMAs with the central DMA (Fig. 5), which helps in mining the data from entire manufacturing processes. The data relating to each process and its adjacent processes are mined independently. The central DMA mines data related to different steps of the manufacturing process. Each process will have its own local DMA and data warehouse where the data will be stored and analysed when required. The data can also be transferred to the main data warehouse (if required), which has a direct link with a central DMA for the analysis of the whole system's data. The whole process is thus supported and can be explored by the remainder of the network. The activities and results are consistently communicated to the knowledge pool through the central DMA. In an integrated manufacturing environment where the product is developed at separate locations or in discrete steps data mining can best be used to control any individual process or step through identification of hidden information in its associated data. The manufacturing knowledge and limitations thus discovered can be utilized for better quality control of future productions.

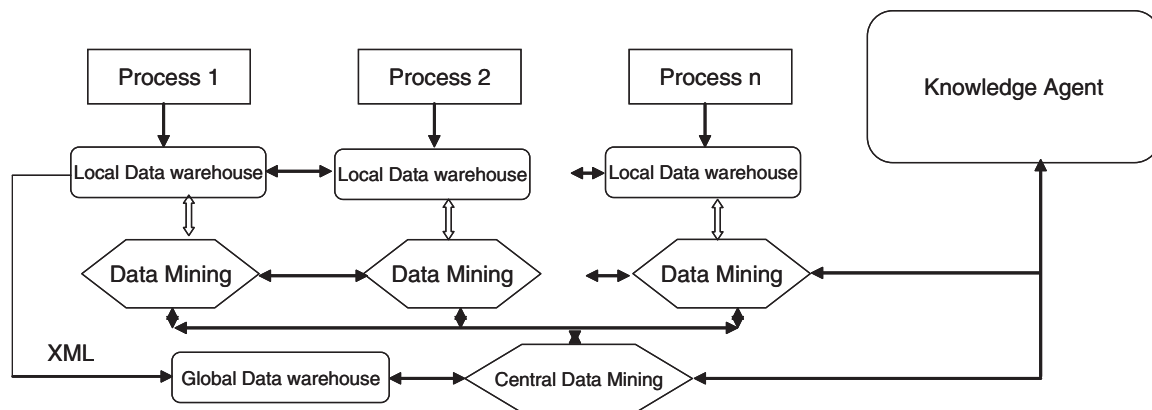


Fig. 5 Integration of data mining agent

The basic idea for integration of DMAs is that rules—principles and concepts applicable to one manufacturing stage—may also be utilized (tested and applied) for other similar stages (by the exchange of activity and rules information via the main data mining engine). This integrated DMA can be applied at a factory level where a product goes into different stages and data for each and every step are collected and stored in a pre-designed data warehouse or in the pattern warehouse as knowledge is much more compact than data. Data mining activities and the main data warehouse will work in parallel during the whole activity with the production process. A standard format is used to build a central data warehouse for an integrated system where production is carried out at different locations. The data from the individual sites will be transferred to the main data warehouse using extensible markup language (XML) format where the data will be mined for the whole process and rules/knowledge extracted will be returned back in the same format.

Integrated data mining may also be productive and enable better knowledge exploitation, for example if different rules are identified for two individual, but similar small production steps. The rules could then be tested to see if they are applicable to both steps and if so, the rules can be shared, and each data mining engine can use its knowledge to refine the ‘best’ one for its particular application. In this way, knowledge can be fed into the main data warehouse, so results can be reused in the future. Future applications can then make use of the stored patterns and rules instead of always having to return to the original manufacturing process databases.

3.4 Translator agent

The translator agent manages the communication between decision makers (user agents) and the knowledge agent. Figure 6 shows the different components of this agent. The user interface manages interactions

with the decision maker and the knowledge agent interface manages the knowledge communications with the knowledge agent. The required knowledge to perform these tasks is stored in the agent’s knowledge base. These three components are controlled by the functional module. The translator agent enables the decision makers to view the state of available knowledge, information, and data mining processes. It also interprets the data mining results.

3.5 Knowledge agent

The knowledge agent provides the required knowledge and information of different alternatives and their performance to the decision makers (user agents). Figure 7 represents the architecture of the knowledge agent. It mediates requests from the translator agent interface, analyses these requests with its knowledge base and inference module. It then sifts through the knowledge pool to extract the required information or initiate a DMA to perform the required data mining task. It also mediates the knowledge extracted by the DMA, stores any relevant knowledge in the knowledge pool for future use, and transmits it through the translator agent interface to the decision makers.

3.6 Knowledge pool

The knowledge pool consists of domain knowledge rules and expertise required and generated by the system. Different types of knowledge are integrated in it such as knowledge about the mining process, knowledge about the different operational processes, and the extracted knowledge, such as rules specifying efficient combinations of input parameters for processes or rules for identifying that maintenance interventions are likely soon, etc.. The knowledge pool is updated whenever the DMA finds new or revised solutions. The exact structure of the knowledge pool, incorporated knowledge management [67, 68],

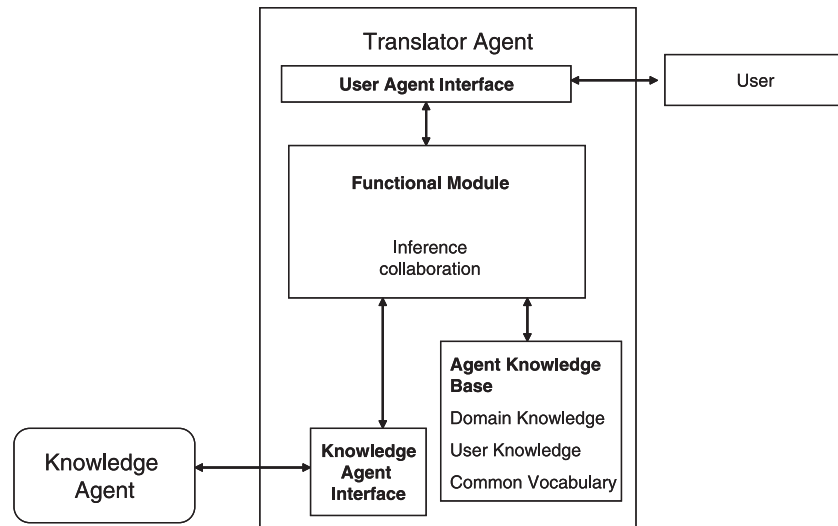


Fig. 6 Translator agent architecture

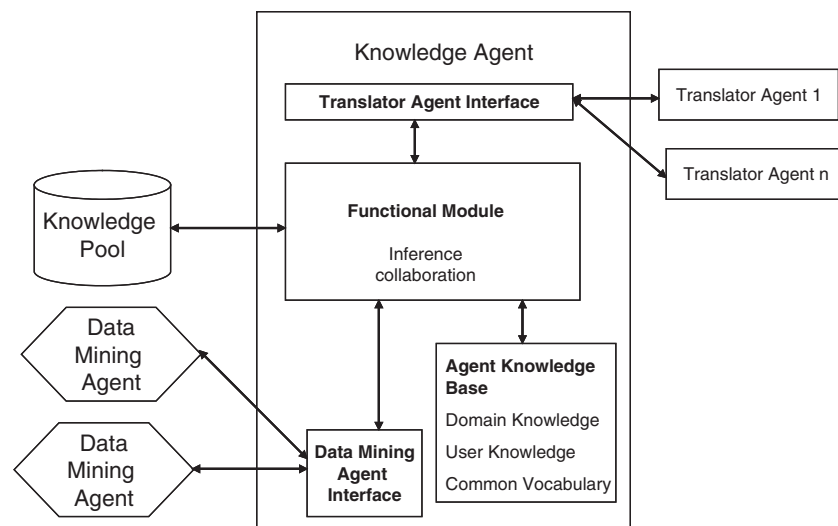


Fig. 7 Knowledge agent architecture

and best-practice identification and capture is beyond the scope of this work. However, these things must be considered in future. In this work, the knowledge pool has been considered to consist simply as a set of rules. The knowledge pool content can include all types of digitized knowledge without format restriction. The knowledge pool consists of procedural knowledge (steps to solve problems), declarative knowledge (descriptions of problems), meta knowledge (knowledge about knowledge), common knowledge (rules of thumb), and structural knowledge (knowledge structures), i.e. all types of knowledge that an organization may use to describe its functioning and its environment. End users would interact with the knowledge pool through client tools,

e.g. Java and Web applications, to insert documents and their associated metadata into the knowledge pool, search for relevant documents, and download them from the knowledge pool. Special care should be taken during development of these tools to ensure that they are adaptable and that changes to the underlying structure of the knowledge pool should be relatively easy to accommodate.

4 PROTOTYPE OF THE SYSTEM

The goal of SFC is to achieve the best possible correspondences between external demands and internal possibilities of the enterprise. The necessary

functions can be summed up as 'logistic process chain' and 'technological process chains' forming the basis of production [69]. Control represents the implementation of planning instructions, in the form of value-adding processes.

The planning and scheduling process begins when a customer order is released. This order will be in the form of similar structure or parts and then the process planner agent generates a linked list of feature-based operations. In addition to the process plan alternatives, the system will also have knowledge of each job's due date. When a job is released to the scheduler agent, a job agent is created to represent the physical order and given the process planning and due date information for storage in its own database. A resource agent exists for each system resource and possesses knowledge of its capability and cost in terms of the feature it can produce, the part shape and size it can hold (fixtures), as well as the transporting devices it needs or possesses. The scheduler agent then generates an optimal plan based on information provided by the job and resource agent. The shop floor supervisor agent receives the information about the optimal sequence along with different process plans and other constraints on the task. Its main function is to closely follow the sequence generated by the scheduling agent. Whenever an urgent order enters the system, it notifies its advent to the shop floor supervisor. The shop floor supervisor then re-sequences the jobs depending upon the scenario.

In this section, the prototype of the proposed decision support system is presented. The objective is to provide the user agents with alternatives for decision making in different scenarios through agent interactions (using the knowledge agent). Each agent is

provided with a set of objectives to meet and has a constraint to fulfil. These agents would need information and knowledge to meet these goals in the ever-changing environment. These agents will communicate with the decision support system for the required information and knowledge. The proposed system functioning has been represented in Fig. 8.

Each user agent (e.g. shop floor supervisor agent) is provided with a set of goals/targets (e.g. product dimension) to achieve by the higher level of planning in the system. The agent aims to achieve those goals while fulfilling its constraints. The user agent needs to cope with changes in the dynamics of the shop floor and take optimal decisions. The decision support system helps the user agent by providing relevant information and knowledge. The user agent logs into the translator agent by specifying their goals/problem (e.g. output dimensions are out of tolerance). The translator agent then converts the problem into a suitable format for analysis by the knowledge agent (e.g. control signature for product: range of process variable values that will result in a product within tolerance).

The knowledge agent then searches the knowledge pool to find whether such information and knowledge are available or not. The information and knowledge if found is passed to the user agent through the translator agent (e.g. as a set of rules that include process variables and associated ranges of values for those variables which should produce products with the specified output dimensions in tolerance. The quality of the knowledge provided should also be considered and therefore details of the support and confidence for the provided rules could also be passed to the user agent). If appropriate knowledge is not found

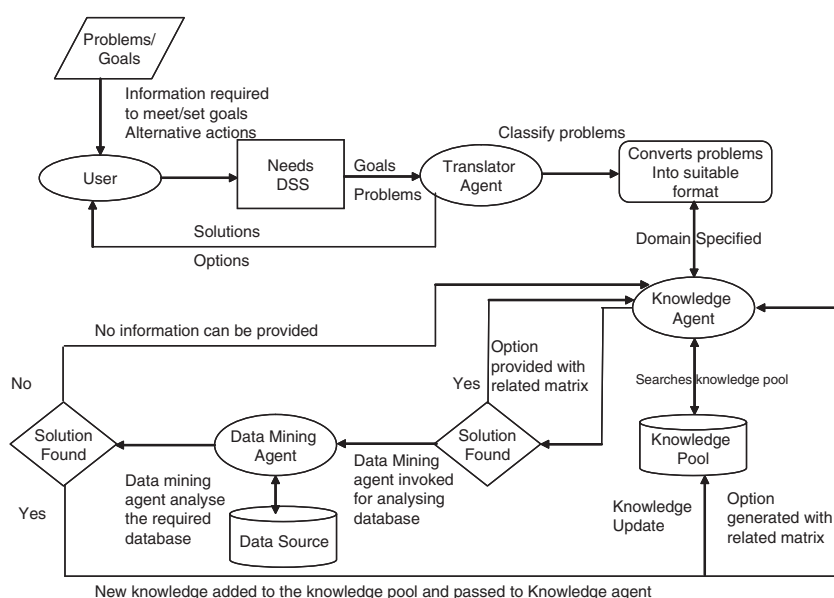


Fig. 8 Proposed system functioning

in the knowledge pool, the DMAs are invoked to analyse the historical data to try to generate suitable information and knowledge. The solution if found is passed to the user agent through a translator agent and is also added to the knowledge pool. Otherwise, a message that no support can be provided is passed to the user agent.

Typically, the decision support system would be invoked when the user agents cannot meet the set of goals or target assigned to them, e.g. if the shop floor supervisor agent is allocated the task to meet a particular product dimension, it might initially control the process input variable to meet this requirement. However, it would need to invoke decision support, whenever the product dimensions are going out of tolerance. This agent would then log into the translator agent with information about the dimension (e.g. length at section) for which information is sought. The translator agent will then seek for a control signature from the knowledge agent. The knowledge agent will search the knowledge pool to see if relevant information exists or if not, will invoke the DMA to generate the result, and will pass the information back to the shop floor supervisor agent.

The following section describes the prototype implementation and its application to the proposed decision support model as presented in the previous sections.

5 PROTOTYPE APPLICATION

This section presents the prototype of the different agents in the system:

5.1 User agent

In this application a process planner agent, job agent, resource agent, scheduler agent, and supervisor agent have been considered as user agents. These agents carry out certain functions in the shop floor and seek information from the decision support system on different scenarios.

The user agent class contains the base functionality of all user agents. The major data elements from the decision application point of view are contained in *wishlist*, which is a vector of data elements representing scenarios on which the user agent is seeking assistance from the decision support system. The *negotiations* are recorded as part of the hashtable of negotiations that are in progress. The *AgentBody* method contains the domain specific tasks for the agent and the *process* method initializes the thread enabling the agent to carry out its functions. This method registers the agent in the decision support system environment if it detects changes in its operations that it cannot handle by itself. The registration

in the decision support method is performed with the help of the *Register* method and it enables the user agent to add scenarios from the wish list. The *Processdsspop* method kicks off the negotiation with the decision support environment by sending a message to the translator agent by invoking *Processdss-agent* method. It processes the message received from the decision support environment with the *Processdssmsg* method.

Figure 9 represents a graphical user-interface (GUI) for this agent when it is initialized to seek assistance. In this application the user agent logs into the translator agent by using three pieces of information: user (identifier), information (type of information required), and side (dimension of interest) to describe the scenarios in the environment. These are just a set of simple elements used in the prototype application as they enable information and mined knowledge from the example in Shahbaz *et al.* [70] to be used in this experimental application. However, for a real-time application more elements need to be added and an adaptive GUI, which can accommodate multiple parameters depending on the particular current user requirements, needs to be incorporated. The lower text box is the space reserved for the information obtained from the decision support environment.

5.2 Translator agent

A translator agent manages the communication between user and knowledge agents. A single instance of translator agent was implemented for experimentation. The translator agent contains two hashtables—one that is a registry for all agents in the environment and one that contains communities that are of interest in the environment. The *Msgcur* attribute handles the current message that is being handled. The *Reset* method clears all the agents and communities from the hashtables. The *initialise* method initializes the translator agent and registers it with the decision support environment. The *StartAgentProcess* method is used to start its thread and make it become runnable when a user agent logs into it. The *GetTaskDescription* method retrieves the scenarios from the user agent and *process* method initiates the thread for this inquiry. The *ProcessEventAgent* method implements the *EventListener* interface and implements the domain code for the previously obtained task description and generates the message to be routed. The *Routemsg* method directs the message to the required agent (user or knowledge agent). It takes *msg* as parameter and uses the reference from hashtable to direct the message. The *Delagent* method removes the agent from the hashtable for which the information has been provided.

Figure 10 presents the GUI for this agent, the upper text field box represents the field of knowledge that is

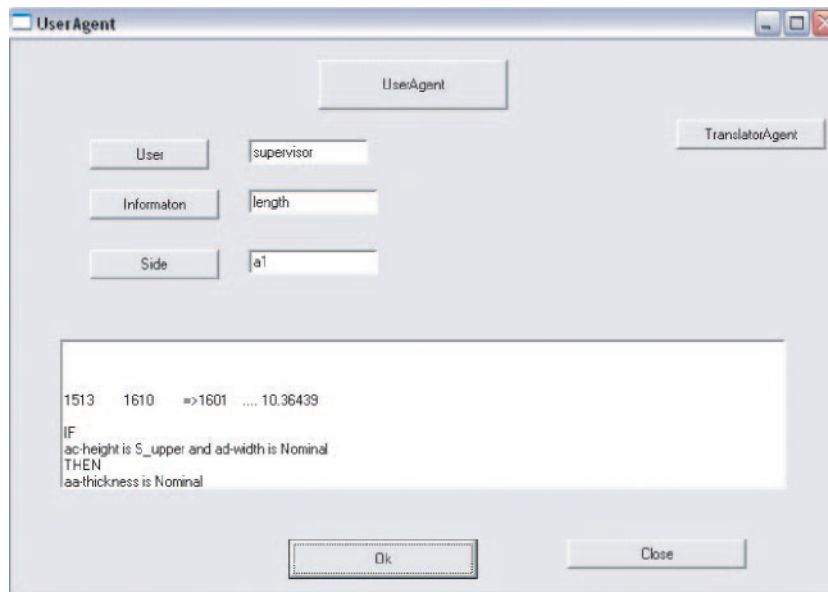


Fig. 9 User agent

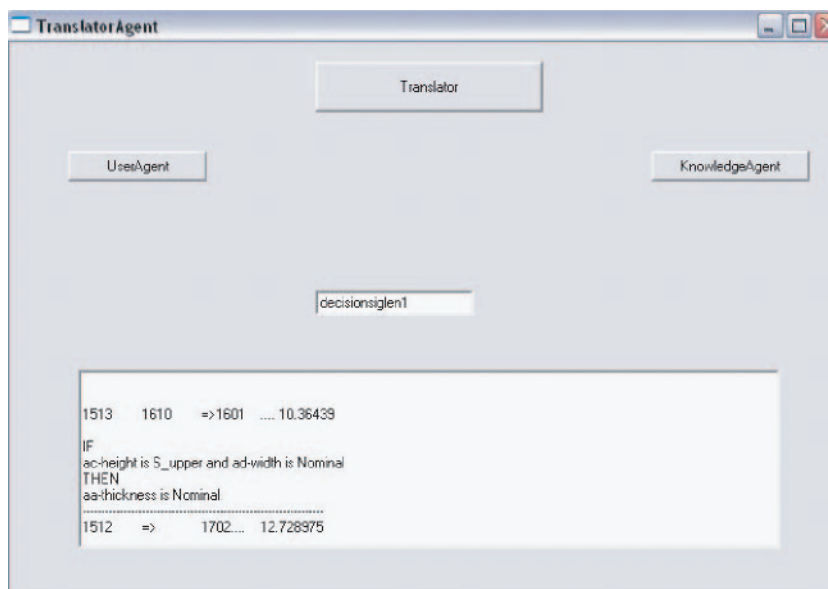


Fig. 10 Translator agent

to be searched, and the lower text box will be filled with the knowledge that will be transferred by the knowledge agent. The field of knowledge in this example is a decision signature obtained in the example shown in Shahbaz *et al.* [70].

5.3 Knowledge agent

A knowledge agent provides the required information in response to the message received from the translator agent. It searches the knowledge stored in the

knowledge pool in correspondence to the information sought by the user agent and if suitable knowledge exists it returns this to the user agent via the translator agent.

The knowledge agent class contains the base functionality necessary for the knowledge agent to provide the user agents with answers to their queries. The major data members of this class include *KnowledgePool*, which is a hashtable of elements that help in searching knowledge pool and *Negotiation* which is a hashtable of negotiations that are in progress.

The *Initialise* method initializes the different data elements in this agent and registers it in the environment. The *register* method is invoked when the translator agent logs into it and starts the agent's thread running. The *TaskDescription* method enables it to retrieve the task which it is required to perform. *ProcessAgentEvent* method searches the knowledge pool for the task description. It passes the information obtained from the knowledge pool using the *Message* method and processes the messages received from other agents through the *ProcessMessage* method. The thread goes to sleep when there are no items in the hashtable and deletes the items from the hashtable for which information has been provided.

5.4 Data mining agent

A DMA provides the required information in response to the message received from the knowledge agent. It searches the databases in correspondence to the information sought by the user agent and applies different data mining algorithms to generate the relevant knowledge. The knowledge thus generated is passed to the knowledge agent and is stored in the knowledge pool.

The DMA class contains the base functionality necessary for the DMA to provide the knowledge agents with answers to their queries and update the knowledge pool. The major data members of this class include *KnowledgePool*, which is a hashtable of elements that help in updating the knowledge pool and *search* which is a hashtable of the information searches that are in progress. The *Initialise* method initializes the different data elements in this agent and registers it in the environment. The *register* method gets invoked when the translator agent logs into it and starts the agent's thread running. The *TaskDescription* method enables it to retrieve the task which it is required to perform. The *ProcessAgentEvent* method searches the database for knowledge generated for the obtained task description. It passes the information generated by the data mining task to the knowledge agent and updates the knowledge pool by using the *Message* method and processes the messages received from the other agents through the *ProcessMessage* method. The thread goes to sleep when there are no items in the hashtable and deletes the items from the hashtable for which information has been provided.

A prototype implementation was built using a java application to represent the behaviour of the proposed system where user, translator, knowledge agent, and DMA interact. The applications were constructed using Borland JBuilder 3.0 interactive environment. The visual builder enabled the complete GUI to be created by using swing components in a drag-and-drop style and automatically generate

codes for creating GUI controls and event handlers. The logic for other agents was then added to set up the decision support infrastructure.

6 CONCLUSION

The increasing use of computers at different levels in manufacturing enterprises, coupled with advances in information technology and decreasing costs of data analysis, have enabled enterprises easily to store and maintain large volumes of data. These data sources can be valuable assets and are potentially important sources to explore and generate knowledge and information about the system behaviour. Therefore, the manufacturing operation should not only be viewed as a production centre but can also be considered as a nucleus of information and knowledge for performance improvement. The review of SFC systems revealed that the control architecture must address issues of adaptability, generic applicability, efficient and effective knowledge management and discovery, reactive scheduling and process planning, and have dynamic structure. The intelligent shop floor controller for a dynamic environment must also incorporate mechanisms to monitor the environment in real time, support learning mechanisms, and present relevant information and alternatives to decision makers and assist in making better decisions. These features indicated that improved knowledge management and ongoing learning (through knowledge discovery and reuse) were needed to improve the effectiveness and efficiency of SFC systems. Therefore, this work presents a data-supported manufacturing SFC system, which can benefit from its historical or legacy systems, as well as from its current databases. The proposed agent-based system for decision support in SFC integrates the data mining capabilities to address the issue of knowledge acquisition and support learning mechanism. The decision support system provides different alternatives to the decision maker either by shifting through the knowledge pool or by invoking the DMA.

The proposed system makes use of quasi-heterarchical structures to enhance the decision-making process. The intelligent decision support tool presented in this work incorporates data mining and intelligent agent technology for providing useful information and knowledge. These techniques provide a mechanism for learning in the system. It also provides an approach for integration of data mining processes for the generation of required knowledge and information for different activities on the shop floor into a decision support framework by means of applying intelligent agent technology. An approach for linking different DMAs situated at different

process sites has also been discussed. This linkage helps in transferring the knowledge generated at one site to be reused at other. The proposed data mining enabled decision support tool provides feedback to different levels in a formalized way so that discovered knowledge can be exploited and reused in various ways in the future. The proposed data mining enabled decision support system would thus assist the decision makers in making decisions by providing them with more alternatives with their implications.

Two different approaches have been made to test and demonstrate the proposed system. A partial implementation has been coded to demonstrate the application of an MAS in this context. This research presents the conceptual description and explanation of the set of modules that would be required to build the system and in order to maintain the generic applicability of the proposed approach. The partial implementation has been kept flexible by using JAVA code rather than by using any particular agent environments which would constrain the required detailed specification of hardware and software.

The proposed data mining enabled decision support tool can provide feedback to different levels in a formalized way so that discovered knowledge can be exploited and reused in various ways in the future. One of the important attributes of this approach is that they have the potential for their knowledge to be continuously updated and provide a way for generating knowledge from operational databases. The presented data mining enabled architecture can complement the decision-making process in an ERP system by analysing the data and presenting the different knowledge embedded in the system and help to learn not only what has happened in the operations, but also why it has happened. The data mining enabled manufacturing system can therefore help to enhance the analysis and prediction capability of the current enterprise.

It is also important to note that the blind application of data mining to generate knowledge can be dangerous, leading to the use of meaningless patterns. It is therefore always desirable to incorporate prior knowledge and properly to interpret mined patterns. The successful methodology of data mining application should only be incorporated in the decision support system. Additional verification of new knowledge by domain experts is therefore also recommended. Knowledge organization will be an important issue in the functioning of this decision support system. A functional specification for knowledge management should be incorporated which generates a synergy between knowledge generation, operationalization, and support quicker access to it.

REFERENCES

- 1 **Cho, H.** and **Wysk, R. A.** Intelligent workstation controller for computer integrated manufacturing: problems and models. *J. Mfg System*, 1995, **14**(4), 252–263.
- 2 **Lima, R. M., Sousa, R. M., and Martins, P. J.** Distribute production planning and control agent based system. *Int. J. Prod. Res.*, 2006, **44**(18–19), 3693–3709.
- 3 **Jones, A. T.** and **McLean, R. C.** A proposed hierarchical control model for automated manufacturing systems. *J. Mfg Systems*, 1986, **5**(1), 15–25.
- 4 **Yoon, H. J.** and **Shen, W.** Simulation based real time decision making for manufacturing automation system: a review. *Int. J. Mfg Technol. Mgmt*, 2006, **8**(1–3), 188–202.
- 5 **Pinedo, M.** *Scheduling-theory, algorithm and systems*, 1995 (Prentice-Hall, New Jersey).
- 6 **Park, S., Jang, P. Y., and Cho, H.** Petri-net based rapid development of a task execution module of equipment controller for distributed shop floor control. *Computers in Industries*, 2001, **45**, 155–175.
- 7 **Son, Y. J.** and **Wysk, R. A.** Automatic simulation model generation for simulation-based, real-time shop floor control. *Computers in Industries*, 2001, **45**, 291–308.
- 8 **Shin, J.** and **Cho, H.** Rapid development of a distributed shop floor control system from an XML model-based control software specification. *Int. J. Prod. Res.*, 2006, **44**(2), 329–350.
- 9 **Morton, T. E.** and **Pentico, D.** *Heuristic scheduling system*, 1993 (Wiley, New York).
- 10 **Heragu, S. S., Graves, R. J., Kim, B. I., and Onge, A. S.** Intelligent agent based framework for manufacturing systems control. *IEEE Trans. Systems, Man and Cybernetics – Part A: Systems and Humans*, 2002, **32**(5), 560–573.
- 11 **Eberts, R. E.** and **Nof, S. Y.** Distributed planning of collaborative production. *Int. J. Advd Mfg Technol.*, 1993, **8**, 57–71.
- 12 **Chen, R. S., Lu, K. Y., Yu, S. C., Tzeng, H. W., and Chang, C. C.** A case study in design of BTO/CTO shop floor control system. *Inf. Mgmt*, 2003, **41**, 23–37.
- 13 **Shiue, Y.-R.** and **Guh, R.-S.** The optimization of attribute selection in decision tree-based production control. *Int. J. Advd Mfg Technol.*, 2006, **28**, 737–746.
- 14 **Hong, J., Prabhu, V., and Wysk, R. A.** Real-time batch sequencing using arrival time control algorithm. *Int. J. Prod. Res.*, 2001, **39**(17), 3863–3880.
- 15 **Gao, Q., Luo, X., and Yang, S.** Stigmergic cooperation mechanism for shop floor system. *Int. J. Advd Mfg Technol.*, 2005, **25**, 743–753.
- 16 **Srinivas, Tiwari, M. K., and Allada, V.** Solving the machine-loading problem in a flexible manufacturing system using a combinatorial auction-based approach. *Int. J. Prod. Res.*, 2003, **42**(9), 1879–1893.
- 17 **Wang, H.** Intelligent agent-assisted decision support systems: integration of knowledge discovery. *Expert Systems with Applic.*, 1997, **12**(3), 323–336.
- 18 **Ozbayrak, M.** and **Bell, R.** A knowledge based decision support system for the management of parts and tools in FMS. *Decision Support Systems*, 2003, **35**, 487–515.
- 19 **Cunha, C. D., Agard, B., and Kusiak, A.** Data mining for improvement of product quality. *Int. J. Prod. Res.*, 2006, **44**(18), 4027–4041.

- 20 Jin, Y. and Ishino, Y. DAKA: design activity knowledge acquisition through data mining. *Int. J. Prod. Res.*, 2006, **44**(15), 2813–2837.
- 21 Kusiak, A. A data mining approach for generation of control signatures. *J. Mfg Sci. Engng*, 2002, **124**(4), 923–926.
- 22 Lau, H. C. W., Ning, A., Pun, K. F., Chin, K. S., and Ip, W. H. A knowledge based system to support procurement decision. *J. Knowledge Mgmt*, 2005, **9**(1), 87–100.
- 23 Tiwana, A. *The knowledge management toolkit*, 2000 (Prentice-Hall, Englewood Cliffs, NJ).
- 24 Harding, J. A., Shahbaz, M., Srinivas, S., and Kusiak, A. Data mining in manufacturing: a review. *Trans. ASME, J. Mfg Sci. Engng*, 2006, **128**, 969–976.
- 25 Shiue, Y. R. and Su, C. T. An enhanced knowledge representation for decision tree based learning adaptive scheduling. *Int. J. Computer Integr. Mfg*, 2003, **16**(1), 48–60.
- 26 Sha, D. Y. and Liu, C. H. Using data mining for due date assignment in a dynamic job shop environment. *Int. J. Adv. Mfg Technol.*, 2005, **25**, 1164–1174.
- 27 Koonce, D. A. and Tsai, S. C. Using data mining to find patterns in genetic algorithm solutions to a job shop schedule. *Computers Ind. Engng*, 2000, **38**, 361–374.
- 28 Bauer, A., Browen, R., Browne, J., Duggan, J., and Lyons, G. Shop floor control systems: from design to implementation, 1991 (Chapman and Hall, UK).
- 29 Melnyk, S. A. and Carter, P. L. Production activity control (Ed. R. D. Irwin), 1987 (Homewood, IL).
- 30 Dilts, D. M., Boyd, N. P., and Whorms, H. H. The evolution of control architectures for automated manufacturing system. *J. Mfg Systems*, 1991, **10**(1), 79–93.
- 31 Jones, A. T. and Saleh, A. A proposed hierarchical control architecture for automated manufacturing system. *J. Mfg Systems*, 1986, **5**(1), 15–25.
- 32 Simpson, J. A., Hocken, R. J., and Albus, J. S. The automated manufacturing research facility of the national bureau of standards. *J. Mfg Systems*, 1982, **1**(1), 17–31.
- 33 Duffie, N. A. and Prabhu, V. V. Real-time distributed scheduling of heterarchical manufacturing systems. *J. Mfg Systems*, 1994, **13**(2), 94–107.
- 34 Hatvany, J. Intelligence and cooperation in heterarchical manufacturing systems. *Robotics Comp. Int. Mfg*, 1985, **2**(2), 101–104.
- 35 Nwana, H. S., Lee, L., and Jennings, N. R. Co-ordination in multi-agent systems. In *Software agents and soft computing* (Eds H. S. Nwana and N. Azarmi), 1997, pp. 42–58 (Springer, New York).
- 36 MacChiaroli, R. and Riemma, S. A negotiation scheme for autonomous agents in job shop scheduling. *Int. J. Computer Integr. Mfg*, 2002, **15**(3), 222–232.
- 37 Dewan, P. and Joshi, S. B. Implementation of an auction based distributed scheduling model for a dynamic job shop environment. *Int. J. Computer Integr. Mfg*, 2001, **14**(5), 446–456.
- 38 Veeramani, D., Wang, K. J., and Rojas, J. Modelling and simulation of auction based shop floor control using parallel computing. *IIE Trans.*, 1998, **30**, 773–783.
- 39 Maturana, F., Shen, W., and Norrie, D. H. Metamorph: an adaptive agent-based architecture for intelligent manufacturing. *Int. J. Prod. Res.*, 1999, **37**(10), 2159–2173.
- 40 McFarlane, D. G. and Bussmann, S. Developments in holonic planning and control. *Production Planning Control*, 2000, **11**(6), 522–536.
- 41 Shen, W., Lang, Y. T., and Wang, L. iShopFloor: an internet-enabled agent-based intelligent shop floor. *IEEE Trans. Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 2005, **35**(3), 371–381.
- 42 Lea, B. R., Gupta, M. C., and Yu, W. B. A prototype multi-agent ERP system: an integrated architecture and a conceptual framework. *Technovation*, 2005, **25**, 433–441.
- 43 Bongaerts, L., Brussel, H. V., and Valckenaers, P. Scheduling execution using perturbation analysis. In *Proceedings of 1998 IEEE International Conference on Robotics and automation*, 1998. Leuven, Belgium.
- 44 Wullink, G., Gademann, A. J. R. M., Hans, E. W., and van Harten, A. Scenario-based approach for flexible resource loading under uncertainty. *Int. J. Prod. Res.*, 2004, **42**(24), 5079–5098.
- 45 Roy, D., Anciaux, D., and Vernadat, F. SYROCO: a novel multi-agent shop floor control system. *J. Intelligent Mfg*, 2001, **12**(3), 295–308.
- 46 Lastra, J. L. M. and Colombo, A. W. Engineering framework for agent based manufacturing control. *Engng Applic. Artificial Intell.*, 2006, **19**, 625–640.
- 47 Lin, G. Y. and Solberg, J. J. Integrated shop-floor using autonomous agents. *IIE Trans.*, 1992, **24**(3), 57–71.
- 48 Lin, G. Y. and Solberg, J. J. Autonomous control for open manufacturing systems. In *Computer control of flexible manufacturing systems* (Eds S. B. Joshi and J.S. Smith), 1994, pp. 169–205 (Chapman & Hall, London).
- 49 Tilley, K. J. Machining task allocation in discrete manufacturing systems. In *Market-based control: a paradigm for distributed resource allocations* (Eds S. H. Clearwater), 1996, pp. 225–252 (World Scientific, River Edge, NJ).
- 50 Metaxiotis, K. S., Askounis, D., and Psarras, J. Expert systems in production planning and scheduling: a state-of-the-art survey. *J. Intell. Mfg*, 2002, **13**(4), 253–260.
- 51 Iwamura, K., Okubo, N., Tanimizu, Y., and Sugimura, N. Real-time scheduling for holonic manufacturing system based on estimation of failure status. *Int. J. Prod. Res.*, 2006, **44**(18–19), 3657–3675.
- 52 Sun, Y. L., Chang, T. M., and Yih, Y. Learning based adaptive controller for dynamic manufacturing cells. *Int. J. Prod. Res.*, 2005, **43**(14, 15), 3011–3025.
- 53 Shen, W. and Norrie, D. H. An agent-based approach for information and knowledge sharing in manufacturing enterprise networks. *Int. J. Networking and Virtual Organisations*, 2004, **2**(2), 173–190.
- 54 Pham, D. T. and Afify, A. A. Machine-learning techniques and their applications in manufacturing. *Proc. IMechE, Part B: J. Engineering Manufacture*, 2005, **219**(B), 395–412.
- 55 Han, J. and Kamber, M. *Data mining: concepts and techniques*, 2001 (Morgan Kaufmann Publishers, USA).
- 56 Malkoff, D. B. A framework for real-time fault detection and diagnosis using temporal data. *Artif. Intell. Engng*, 1987, **2**(2), 97–111.
- 57 Ramamoorthy, C. V. and Wah, B. W. Knowledge and data engineering. *IEEE Trans. Knowledge Data Engng*, 1989, **1**(1), 9–16.

- 58 Babiceanu, R. F. and Chen, F. F. Development and application of holonic manufacturing systems: a survey. *J. Intell. Mfg*, 2006, **17**, 111–131.
- 59 Maturana, F. and Norrie, D. H. Multi-agent mediator architecture for distributed manufacturing. *J. Intell. Mfg*, 1996, **7**, 257–270.
- 60 Gou, L., Luh, P. B., and Kyoya, Y. Holonic manufacturing scheduling: architecture, cooperation mechanism and implementation. *Computers in Industries*, 1998, **37**, 213–231.
- 61 Lee, J. H. and Park, S. C. Agent and data mining based decision support system and its adaptation to a new customer-centric electronic commerce. *Expert Systems with Applications*, 2003, **25**, 619–635.
- 62 Davenport, T. H. and Prusak, I. *Working knowledge: how organizations manage what they know*, 2000 (Harvard Business School Press, Boston, MA).
- 63 Alazmi, M. and Zairi, M. Knowledge management critical success factors. *Total Quality Mgmt Bus. Excellence*, 2003, **14**(2), 199–204.
- 64 Guo, H., Hou, W.-C., Yan, F., and Zhu, Q. A Monte Carlo sampling method for drawing representative samples from large databases. In *Proceedings of the 16th International Conference on Scientific and statistical database management*, 2004 (Greece).
- 65 Maki, H. and Teranishi, Y. Development of automated data mining system for quality control in manufacturing. *Lecture Notes in Computer Science*, 2001, **2114**, 93–100 (Springer-Verlag, Berlin).
- 66 Lin, H. K., Harding, J. A., and Teoh, P. C. An inter-enterprise semantic web system to support information autonomy and conflict moderation. *Proc. IMechE, Part B: J. Engineering Manufacture*, 2005, **219**(B12), 903–911.
- 67 Young, R. I. Informing decision makers in product design and manufacture. *Int. J. Computer Integr. Mfg*, 2003, **16**(6), 428–438.
- 68 Gunendran, A. G. and Young, R. I. An information and knowledge framework for multi perspective design and manufacture. *Int. J. Computer Integr. Mfg*, 2006, **19**(4), 326–338.
- 69 Scherer, E. *Shop floor control—a system perspective from deterministic models towards agile operations management*, 1998 (Springer Verlag).
- 70 Shahbaz, M., Srinivas, Harding, J. A., and Turner, M. Product design and manufacturing process improvement using association rules. *Proc. IMechE Part B: J. Engineering Manufacture*, 2006, **220**(B2), 243–354.