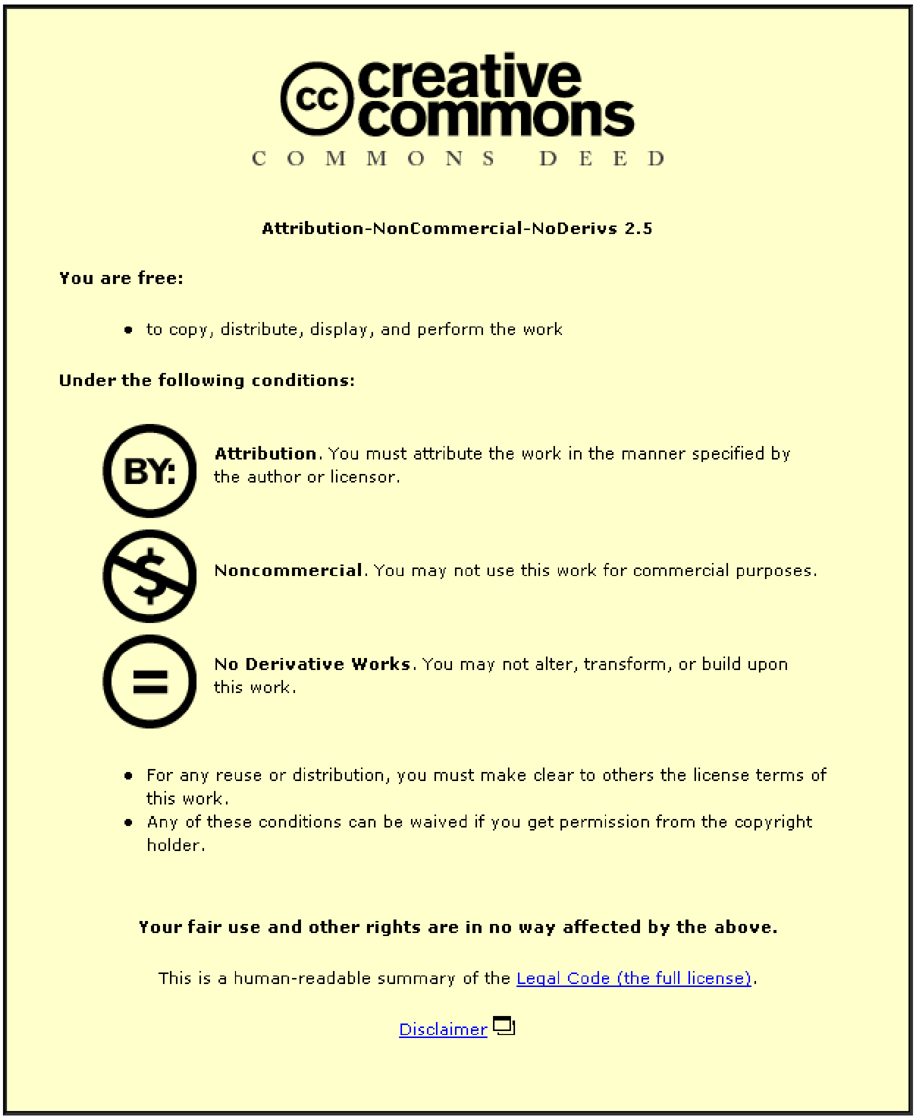
This item was submitted to Loughborough University as an MPhil thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.



creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/

LOUGHBOROUGH
UNIVERSITY OF TECHNOLOGY
LIBRARY

AUTHOR/FILING TITLE

BENNETT, S.C.

ACCESSION/COPY NO

040060611

VOL. NO	CLASS MARK	
- 8 OCT 1993	LOAN COPY	

0400606119

BADMINTON PRESS
18 THE HALFCROFT
SYSTON
LEICESTER LE7 8LD
ENGLAND
TEL. 0533 602917

The Design and Development of an Information Technology Application for Community Transport: A Case Study Approach

by

Sımon C. Bennett

A Master's Thesis

Submitted in partial fulfillment of the requirements for the award of
M. Phil. of the Loughborough University of Technology

September 1992

© by Simon C. Bennett 1992

Mar 93

04006011

W9920920

Abstract

The aim of the research has been to develop a single modular software package for all modes of Community Transport (CT) operations. In order to achieve this aim, the research has set out to examine CT operations, the scope for computerisation of those operations and existing software for them, to analyse the requirements of CT operators through a collaborative process, to develop a data model which supports their operations, and to implement a software package based on this model which provides both a data management system and operational functions.

The thesis defines and describes the various modes of CT operations, surveys existing software for CT operations and examines the scope for computerisation and the problems facing operators which computerisation may be expected to tackle.

The research has been conducted through collaboration with a number of CT operators, both directly and through a user group, in order to establish the data and functional requirements of operators By adopting this participative approach throughout the stages of analysis, design and implementation, it was intended to ensure that the package met the requirements of operators.

A data model for CT, incorporating all modes of CT operations has been developed using entity-relationship analysis, and a core of data common to all modes of operation established. It is argued that this careful analysis of data requirements for all modes of operation provides a more secure base for future developments than is provided by other existing software for single modes of operation.

Four operational functions which go beyond mere data management have been explored and solutions implemented which have been designed to support human users rather than to replace their skills. These are a diary system, vehicle brokerage, routeing and dial-a-ride scheduling

The thesis concludes by suggesting that by establishing a data model which reflects the working of all modes of CT operations, further functionality, in terms of scheduling and routeing, can be added easily, and by suggesting areas for further research into these areas of functionality.

Table of Contents

Page

List of Figures

List of Tables

Foreword

This research was carried out as part of a research project funded by the Science and Engineering Research Council (SERC) between September 1986 and March 1989. The title of the project was 'Application of a Modular IT System to Community Transport Management and Transport Brokerage Projects' (Ref. GR/E/01317), and its aim was to develop and apply an integrated, modular menu-driven IT package to community transport and transport brokerage projects, capable of application throughout the UK. This project followed on from a previous SERC funded research project (GR/D/24852) which evaluated the practical application of IT in improving the effectiveness of the operation of transport services organised by the voluntary sector, particularly community transport (Gillingwater and Sutton, 1988). This research was carried out by Dr John Sutton, who joined the second research project for the final year.

This thesis is written in the context of the fact that the software developed for the research project is now marketed by a commercial software house, and has been installed in over 30 sites in the UK at the time of writing.

Finally, I would like to express my thanks to all the community transport operators who have collaborated with this research and in particular to Ed Passant, Steve Sears and Jim Corbett.

Simon Bennett

September 1992

Chapter 1
Background

The Community Transport Sector in the UK

The term Community Transport (CT) is used in the UK to refer to a variety of specialised transport services provided by a range of different organisations. There is a national charity called Community Transport which is responsible for the provision of management services and direction to a number of local organisations, and Community Transport is the term chosen by the majority of organisations which are not part of the national charity as part of their names and forms part of the titles of a number of other umbrella or advisory organisations, such as the National Advisory Unit for Community Transport (NAUCT), the Community Transport Association (CTA) and Community Transport Magazine (CTM).

In the context of this thesis, CT will be used as generic term to refer to the sector and the range of organisations which fall within it. In the US and Canada, the term paratransit is widely used and will be used here in the context of North American services and developments.

A number of features characterise CT

CT operators provide one or more of the following types of transport service.

1. Dial-a-ride. Dial-a-ride offers a demand-responsive, many-to-many service for people with mobility handicaps, principally the elderly and disabled. Most services of this type require that users are registered as members and that they book in advance, usually by telephone, and many services restrict booking to the day immediately before the service is required.

2. Dial-a-bus. Dial-a-bus offers a many-to-one, semi-scheduled service for people with mobility handicaps. Services usually run to town or city centres or to local or out of town shopping centres. There is a growing interest in such services, and their use to increase the utilisation of vehicle capacity where passengers of dial-a-ride services have common destinations.

3. Social car schemes Social car schemes use volunteer drivers, generally in their own cars, to provide a many-to-many service for the mobility handicapped. The requirement to book in advance is greater than for dial-a-ride, as the operator acts as a broker between passengers' requests for transport and the availability of volunteers.

4. Group hire. Group hire services provide a minibus hire service for organisations in the voluntary sector. In most cases, some or all of the vehicles are accessible, and they thus provide a service for groups which are exclusively for the mobility handicapped or which count them among their members. In some areas, particularly in London, the term Community Transport is used to refer specifically to the group hire style of operation.

5. Vehicle brokerage Vehicle brokerage is a service provided by some group hire operators. Vehicles which do not belong to the operating organisation but to other voluntary, or sometimes statutory, organisations are made available for hire through the broker The aim is to meet transport needs by increasing the utilisation of vehicles which would otherwise be under-used.

6 Furniture and removals Furniture and removals services are provided by some operators, using vehicles to collect and deliver items of furniture for individuals and organisations, or making vehicles available for hire. In some cases, such services are linked to the storage and repair of furniture, or to the provision of cheap furniture to people receiving state benefits.

7. Passenger brokerage. Passenger brokerage aims to match the needs of individuals for transport with spare capacity on vehicles used in a range of services, particularly in rural areas. It can also apply in a CT operation where more than one kind of passenger service is available, and where an individual passenger may request a trip and the operator allocates it to one of the available services on the basis of cost and availability.

CT operators are usually voluntary or non-statutory organisations managed by a management committee which may include representatives of the statutory bodies which provide funds in the form of grants, such as local authorities and public transport authorities. In many cases, consumers of the services are represented in the management structure, whether they are representatives of member voluntary organisations in a group hire service or representatives of passengers in one of the passenger oriented services.

Many organisations in the CT sector are registered charities, and as such aim to provide services in order to benefit specific groups of people, rather than to generate profits. A feature of such organisations, recognised by Drucker (1989) is their clear sense of mission and the ability to innovate.

These features which characterise CT organisations are based on observation of such organisations and self-definition. In contrast, Sutton (1988) and Nutley (1988) seek to place the passenger oriented services which are part of CT in taxonomies of passenger transport services based on a number of dimensions.

Sutton uses two dimensions The first is based on categories of service characteristics: fixed route service; variable route service; contract hire service; and demand-responsive service The second is a hierarchy of primary public transport, secondary voluntary sector community transport, and tertiary public sector social transport.

Nutley provides a more complex analysis of what he calls Unconventional Modes of transport (UCMs), using criteria such as passenger eligibility, type of destination, routeing and timing, multipurpose operation, community or private operation and source of finance.

While these taxonomies are helpful in placing CT operations in the overall context of passenger transport services, individual CT operators may not be as easy to categorise. Firstly, some provide services such as furniture transport and removals which fall outside the realm of passenger transport, and secondly, individual organisations may provide one or more services in various combinations, similar to Nutley's multipurpose operation

As an example, Birmingham Ring-and-Ride provides an exclusively dial-a-ride service, and there are many other organisations around the country which provide only a dial-a-ride service, particularly in the London boroughs. In contrast, in both Derby and Nottingham, dial-a-ride services are provided by CT operators which also provide group hire services, and in the case of Derby Community Transport a dial-a-bus service. This is also the case in a few London boroughs such as Hillingdon, where joint funding has led to the development of a single organisation providing both dial-a-ride and group hire. Figures 1.1 and 1 2 illustrate the two types of organisational structure.

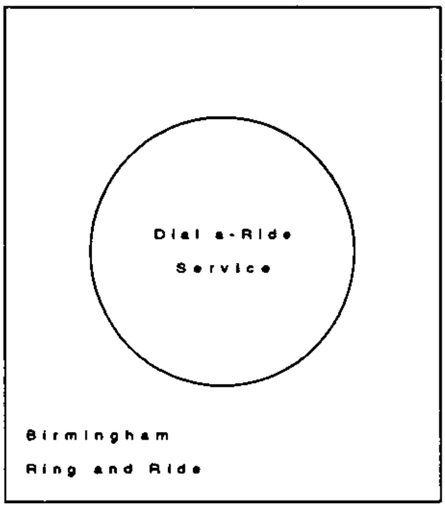


Figure 1.1 *The structure of dial-a-ride in Birmingham*

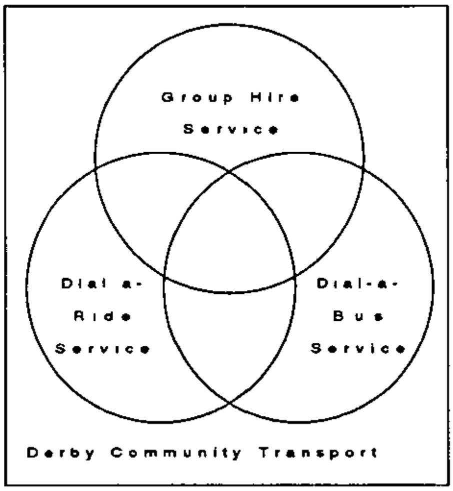


Figure 1.2 *The structure of community transport in Derby*

From the point of view of computerisation of CT services, there are a number of features which are of importance. First among these is the fact that many operators offer multiple services, often using the same vehicles for more than one service. Second is the fact that restrictions on eligibility are usually translated into a requirement that users should be registered with the service as members. Third is the fact that the services are booked in advance, rather than being totally demand-responsive.

The Level of Computerisation in the UK CT Sector

Sutton (1986) carried out a survey of the use of computers in CT agencies in March/April 1986. 600 questionnaire forms were distributed to the readers of Community Transport Magazine 65 organisations responded (11% response rate). Of these, 11 organisations were using computers in the field of transport management or operations, five organisations were using computers in the field of community transport advice, administration or analysis; five organisations planned to implement a computer system in the near future; forty organisations were considering using computers and seeking funding; and four organisations replied that they were not intending to use computers.

When other information available at the time was added to these figures, it emerged that fifteen schemes were using computers in transport management or operations; six agencies were using computers in community transport advice/ administration/analysis; and eleven more schemes were planning to implement a computer system in the near future (Sutton, 1987).

The low level of computer use in community transport operations can be attributed to two factors: firstly, the cost of computer systems and the difficulty for voluntary organisations of attracting funds from funding bodies for the purchase of a computer; and secondly the scarcity of appropriate software packages for CT operations.

The first of these factors is one which affects all voluntary organisations, not just those in the voluntary transport sector. It has changed little, if at all, since the

Wolfenden Committee noted in 1978 that the financial position of most voluntary organisations was less secure than that of statutory organisations (Wolfenden, 1978). Most are and were dependent on a mixture of individual charitable giving, organised fundraising, and grants from charitable trusts, local authorities and other public bodies For many, the high level of capital required to purchase specially adapted vehicles, and the revenue requirements of maintaining vehicles and employing staff, have led to dependence on public funding in the form of local authority grants, the Urban Program and the Inner Cities partnership programme. It is often difficult to persuade funding bodies, whose concern is with the primary service which is to be provided to the public (the provision of transport to the transport disadvantaged), that expenditure on a computer system is justified.

While much of the evidence for this in the CT sector is anecdotal, there is documentary evidence from detailed proposals for computer systems (Wootton Jeffreys, 1980; LOLA, 1985; Beecham,1986) which have never been actioned due to lack of funding.

The second factor is more clearly established. At the time of Sutton's survey, there were only three software packages available for operators of community transport services These are shown in Table 1 1.

Name	Developer	Services
D-A-R	Fletcher Computer Services	Dial-a-ride
HCATS	Hackney Computer and Transport Services	Group hire
RUNBROK-84	Transport and Road Research Laboratory and East Sussex County Council	Passenger brokerage

Table 1.1 *Software packages for CT available in the UK in 1986*

Fletcher's D-A-R

The first of these, Fletcher's Dial-a-Ride (Fletcher Computer Services, 1984) is a commercial package, developed in collaboration with the National Advisory Unit for Community Transport for the Ring-and-Ride operations in Manchester and Birmingham, and subsequently installed also in Haringey (Marsh and Jarratt, 1985). It only provides facilities for the dial-a-ride operations, and some rudimentary accounting functions. It was written to run under the PICK operating system. The cost of the package in 1984/5 is shown in Table 1.2.

Equipment	**Cost**	
	4 Users	**8 Users**
ADDS Mentor Mini Computer		
PICK Operating System		
Modem		
256 Kb Memory		
Printer Port		
M2000	9,995	
15 Mb Disc		
4 VDU Ports		
M2500		13,995
27 Mb Disc		
8 VDU Ports		
Uninterruptible		
Power Supply		
VDU's @£650 each	2,600	5,200
Low speed printer	745	
High speed printer		1,930
Dial-a-Ride Software	2,000	2,000
Total	£15,340	£23,125

Table 1.2 *Cost of Fletcher's D-A-R - 4 and 8 user systems in 1984/5*

Fletcher's Dial-a-Rıde stores records of vehicles, drivers, passengers and passenger trips. Vehicle schedules are maintained manually, either on sheets of paper or on wall-boards, whıle the passenger trips which make up these schedules are recorded on the computer. It is also possible to record refusals, in the situatıon where the operator is unable to meet the passenger's request for transport, and cancellations. Printed trip schedules are produced by sorting the passenger trip file by vehicle and pick-up time for a particular date. A number of printed reports are available from the system. A summary structure of Dial-a-Ride is shown in Figure 3.1, in Chapter 3 where ıts data structure is analysed in more detail.

The objective of the system ıs descnbed as being 'to assist in the ınitıal booking of trip requests, to assist in the clerical operatıons, and to provide analyses to monitor the effectiveness of the system ' (FCS, 1984)

Fletcher's Dial-a-Ride has since been installed ın a number of other dial-a-ride operations, and modificatıons have been made to the standard package at some sites.

HCATS

The second package was developed by Hackney Communıty Transport for its own use, startıng ın 1982, and then offered to other communıty transports at a low cost wıth a view to co-operatıve development (Hackney C&T Systems, 1985). It was originally written to run under the CP/M operating system, and subsequently modıfied to run under Concurrent CP/M or Concurrent DOS.

The Hackney package was designed for communıty transport group hıre servıces only. Provısıon was made subsequently for dıal-a-nde services also. The group hıre suite of the package, as ıt stood in 1985, maıntained records of member groups, drivers, vehicles and invoicing. It also offered some analytical reports on the data held in these fıles

In 1985, ıt was proposed to make the software available to other communıty transport group hıre operators through a non-profit-makıng organisation to be set up ın conȷunctıon with Communıty Transport Services. This organisatıon, Hackney Computer and Transport Services, continued to develop the software for Hackney Communıty Transport, and became more commercıal, sellıng complete computer systems includıng software.

The price of a system, published in 1988 is shown in Table 1.3.

Equipment	**Cost**	
	1 User	**4 Users**
Future Computers FX31	2,995	2,995
Concurrent DOS		
30 Mb Disc		
1Mb Memory		
800 Kb Floppy Drive		
Future Computers FX20 @ £1,495		4,485
Concurrent DOS		
512 Kb Memory		
800 Kb Floppy Drive		
Dot Matrix Printer	445	445
Community Transport Suite	199	199
Dial-a-Ride Suite	199	199
Multi-user Option		850
Total	£3,838	£9,173

Table 1 3 *Cost of Hackney Computer and Transport Services computer system and software - 1 and 4 user systems in 1988*

The package has since been modernised and rewritten using dBase, and is sold by F1 Transport Systems (Jones, 1989).

RUNBROK-84

The third package was developed for East Sussex County Council in collaboration with the Transport and Road Research Laboratory (TRRL). It only caters for passenger brokerage using public transport It was originally written in FORTRAN to run on an ICL mainframe using the VME operating system and subsequently adapted to run on an IBM compatible PC under MSDOS.

This package is now marketed as TDD-ESCORT by Transport Design and Development (TDD). It has never been used within the community transport sector, and is aimed at the local authority market, but is included because it provides for a service whch many see as falling within the remit of CT. It is described as being designed 'to assist local authorities (and other providers of passenger transport services) in the tasks of booking and managing operations, and thus to achieve maximum day-to-day operational efficiency and cost effectiveness.' (Transport Design and Development, Undated).

The package maintains records of scheduled transport services on a zonal basis, passengers and passenger journeys. Requests for passenger journeys are matched to suitable services passing through the appropriate zones. Daily operational schedules are produced, and analytical reporting facilities are available.

Cost figures are not easily available for this package, it is sold as part of a consultancy service, and the price depends on the customer and the nature and size of the operation.

The significant factor about each of these packages, is that at the time of Sutton's survey, they only addressed the requirements of one type of service, and only one has subsequently been developed to cater for another type of service also.

Other developments

Given the lack of readily available commercial software packages suitable for community transport operators, particularly those operating more than one type of service, other community transport operators developed their own software packages in-house Apart from Hackney Community Transport which began its development in this way, examples of this approach to the problem where the development of software had begun at the time of Sutton's survey are Derby, Ealing and Edinburgh Community Transports. In each case, the increasing availability and reducing price of microcomputers and off-the-peg software, which has characterised small business computing in the 1980's, was combined with the availability both of suitable funding and of professional computer expertise either within the organisation in the person of one or more volunteers, or available at minimal cost to the organisation.

In each case, the CT organisation provides more than one service. Derby provides group hire, dial-a-ride and dial-a-bus, and both Ealing and Edinburgh provide both group hire and furniture transport. In each case, software was purchased to allow the computer to be used for word-processing, and to provide a facility for computerised invoicing from vehicle log-sheets. Database or spreadsheet packages were used to achieve the second of these. Derby used Delta, a database; Ealing used Symphony, an integrated spreadsheet package; and Edinburgh used dBase, probably the most commonly used microcomputer database.

The Derby experience is probably the best documented of the three (Ford and Forkin, 1987; Ford, 1988), while the experience of Edinburgh has been covered in Community Transport Magazine (Jones, 1988).

Since the start of this research, two further packages have become available: the first as a result of development by a CT operator, and the second as a result of this research. The first of these is the Dial-a-Ride Scheduler developed for Bolton Dial-a-Ride, and marketed by Blue Collar Software (Williamson, 1987). This package has been written in dBase to run on an MSDOS compatible microcomputer, and provides facilities for dial-a-ride operators to take trip bookings and include them in a vehicle schedule The second is the package MULTI TRIP developed by the IT in CT Project at Loughborough University, which is the subject of this research. Table 1.4 shows the software packages available in 1989 for CT operators and an approximate number of installations of each.

Name	Developer/Distributor	Services	Installed
D-A-R	Fletcher Computer Services	Dial-a-ride	10
F1 (HCATS)	F1 Transport Systems (Hackney CT)	Group hire Dial-a-ride	5
TDD-ESCORT	Transport Design and Development	Passenger brokerage	1
Dial-a-Ride Scheduler	Blue Collar Software	Dial-a-ride	2
MULTI TRIP	Wise Software Ltd (IT in CT Project)	Group hire Vehicle brokerage Dial-a-ride	20

Table 1.4 *Software packages for CT available in the UK in 1989*

The Situation in North America

The situation in the UK contrasts sharply with that in North America. The Urban Mass Transportation Administration (UMTA, 1984) lists 24 microcomputer software packages for paratransit planning and operations either in use or in development in the US in 1984, while Geehan (1985) lists 6 programs in use in Canada in 1985. The American journal 'Time Capsule' lists 10 packages specifically for paratransit scheduling and despatching (excluding planning and forecasting), available in North America in late 1985 (Bower, 1985). Details of three North American packages are included here as examples for comparison with what was available in the UK at the time.

The first of these is TRANSIT, or Special Services Transit (SST), developed as a demonstration program by the UMTA and the Transportation Systems Center for paratransit management. It was designed to demonstrate 'an approach to automating the reservation and reporting data management functions essential to the operation of a small specialized (elderly and handicapped) transit property' (Giangrande, 1985). It is available through the Transit Industry Microcomputer Exchange (TIME) Support Center in a version which can be used to manage a full-scale paratransit operation or as a demonstration version.

TRANSIT was originally written in dBASE II, an industry standard database management system for IBM PCDOS/MSDOS compatible microcomputers. It was subsequently rewritten in dBASE III as SST3 (Time Capsule, 1988) by the Texas Transportation Institute of Texas A&M University. As originally written, it was designed to run on an IBM PC/XT compatible microcomputer with a hard disc drive, and an IBM or EPSON compatible printer. As such, it is a single user system, allowing the use of the software and data by a single operator, dedicating the computer to taking reservations.

TRANSIT is exclusively for the operation of dial-a-ride type paratransit services, as are all the North American packages covered here. It maintains data on clients, vehicles and funding sources, and handles trip bookings and standing client reservations which are inserted into vehicle schedules chosen by the operator. Trip schedules are printed out using the space-filling curve algorithm (Bartholdi et al., 1983) described in Chapter 4. Reports on client demographics and vehicle usage are produced by the package.

TRANSIT is available for a nominal charge from the TIME Support Center, although it is provided in the expectation that it will be customised by the end-user familiar with dBASE.

The second package is Dial-A-Ride Transit Manager (DART Manager), originally developed for BC Transit in Victoria, British Columbia, Canada (Geehan, 1985), and subsequently adopted by BC Transit and by Transport Canada as a demonstration project, and installed in the New Westminster-Burnaby-Coquitlam suburban area east of Vancouver.

The software is written in COBOL and was originally developed on an Onyx multi-user minicomputer in order to take advantage of the multi-user operating system, Oasis. The system consisted of the minicomputer with 128kb of RAM and a 6 7Mb hard disc, with a 10Mb streaming tape cartridge, two terminals and a dot matrix printer The software was written by DIALOG Business Computer Systems in consultation with BC Transit and the operator, Accessible Transportation Alternatives Ltd

The package stores records of vehicles, clients and their standing trips, handles reservations and produces driver trip sheets. Trips are inserted manually into schedules by the operators. It also provides reporting functions on clients and vehicle usage.

The software has since been rewritten to run on computers using other operating systems. It is still written in COBOL, but will run on IBM PC/XT compatibles with 512kb RAM and 10Mb hard disk running PCDOS/MSDOS 2 1 or higher. A multi-user version is available to run on IBM PC/AT compatibles with 1Mb of RAM and 30Mb hard disk running the XENIX operating system An additional package has been added to the range. This is DART Manager Graphics, written in Pascal by the Stanford Research Institute of California for the UMTA, which provides an on-screen graphical display of passenger locations and vehicle trips.

By way of contrast, the third package was written to run in a very different computer environment. The software was written by Systemoid Incorporated for the city of Montreal, Quebec, Canada to run on an IBM 4381 mainframe capable of handling 300 terminals (Geehan, 1985; Hamer, 1986). The mainframe is used by other applications and users, but 15 terminals are in use by Transport Adapté du Québec Metro Inc. (TAQM), the operator of services. The software is written in COBOL, and runs in the complex operating system environment of the mainframe under VSAM, CICS and OS/VS.

The goals of the software development were 'to reduce telephone dwell time (and thereby also reduce telephone congestion), to increase bus productivity, and to produce more timely and complete information concerning operations' (Hamer, 1986). Records are maintained of clients and regular trips, and the system handles scheduling and despatching.

The software is different from the others mentioned in the complexity of its scheduling capabilities. It uses a scheduling algorithm developed for Transport Canada, and a database which includes a distance matrix based on the streets of Montreal and a model of travel times which takes into account congested zones and rush-hour congestion

Areas for Computerisation

There are a number of areas of the administration and operations of community transport which lend themselves to computerisation. Some of these are the standard office automation tasks which are appropriate in the computerisation of any small business, while some are specific to the transport sector and to community transport in particular. Table 1.5 shows which tasks fall into each category. Each of these will be described briefly.

Word processing

Like many small organisations and businesses, CT operators produce a wide range of documents. These can be letters, forms, reports, committee minutes, and all the other kinds of documents which are generated by dealing with employees and the public With limited resources to employ staff, there are clear advantages to using computers to produce these documents

Areas of Computerisation
Appropriate in any small organisation or business
Word processing
Invoicing
Accounts
Specific to transport or community transport
Fleet management
Booking or despatching
Routeing
Scheduling
Performance monitoring

Table 1.5 *Areas of computerisation for community transport*

Word processing has become very common over the last ten years, with word processors replacing typewriters in many offices. In any office, word processing gives certain advantages over traditional means of producing typed documents. These advantages have been covered by many authors in the small business context, and by some in the specific context of community transport (Wyatt and Smerk, 1984) Documents can be entered into the word processor, printed in draft, corrected and reprinted without the need to retype the entire document. Copies of documents can be stored on magnetic media, the hard disk or floppy disks, and if indexed can be easily accessible. Standard letters can be held on disk and used over and over again With the addition of mailmerge facilities, a standard letter can be sent out to many different addressees, with the name, address and salutation being changed and printed appropriately on each letter

There are hundreds of word processing packages available off the shelf for personal computers, offering differing levels of complexity and features for the user, and at a range of prices. This easy availability, and the advantages of word processing over typing probably account for the fact that of the twenty-one CT agencies found by Sutton to be using computers (Sutton, 1987), eighteen (86%) were using computers for word processing. In a similar survey in the US, Paaswell, McKnight and Depa found that the most common uses were word processing, budgeting and recording maintenance functions (Paaswell, McKnight and Depa, 1984)

Invoicing

Different types of CT services are charged for in different ways Dial-a-ride, dial-a-bus and social car scheme travel is usually paid for by the individual passenger at the time of travel. Group hire and furniture transport are usually invoiced to the organisation hiring the transport after the event. In the case of passenger brokerage, and sometimes in the case of dial-a-ride and social car scheme travel, the charges for transport may be invoiced to the passenger or to an agency such as a social services or education department after the event.

The task of invoicing uses source documents such as drivers' work sheets on which mileages have been entered, volunteer drivers' expenses sheets or vehicle log sheets. The work of collating, sorting and processing all this information manually can take days of staff time each month. A process which automates this task has clear benefits to the CT operator.

There are no off the shelf invoicing programs available in the same way that word processing packages are available. However, invoicing can be handled in a variety of ways by off the shelf packages or a combination of them.

Ealing and Derby Community Transports chose two different methods to achieve the automation of their invoicing. Ealing CT used an integrated package based around the Lotus spreadsheet, 'Symphony', to produce their invoices Details of each vehicle hire were entered into a spreadsheet through a form on the screen. A different file was held for each month's work. The spreadsheet was programmed using a 'macro' language to calculate the charges from the mileages together with other miscellaneous charges entered by the user, and to print out invoices including the names and addresses of member organisations which were held in a separate file.

Derby CT on the other hand used a database package, 'Delta', to store records consisting of vehicle log sheet entries. This was programmed to sort all the transactions by the type of service, and for those which involved group hire, to calculate the charges for each trip and to total them for each member organisation and print out an invoice. In addition, it analysed mileages for all other services, and for Derby CT's own use of its vehicles for purposes such as maintenance, and produced statistics based on the useage of vehicles for different types of service and by driver (Ford and Forkin, 1987; Ford, 1988).

Other approaches may be possible using off the shelf packages, for example, maintaining records of hire in a database and merging the data into standard letters using a word processor. The critical thing to note in the context of off the shelf packages is that this type of invoicing is very different from straightforward sales ledger invoicing for products with unit prices and quantities, and does not lend itself to automation using accounts packages

Accounts

Like any small business, a CT operator has to keep accounts, both for the obvious internal purposes of financial planning and control, and for the benefit of funding bodies which require details of how public money is being used. In many cases, the books are kept by staff, while the accounts may be kept by a voluntary treasurer While computerised accounts may help in this process, they can also limit the volunteer treasurer to working on the accounts where the computer is accessible, in the CT office. They may also affect auditing costs, and auditors may require access to the computer at a time when it is being used for other operational purposes.

Off the shelf packages are easily available for all aspects of small business accounting, replacing cash books, sales and purchase ledgers and nominal ledgers. If it is desired to maintain an integrated system of invoicing and accounting, it may be necessary to ensure that the accounts package has an interface to the software used for invoicing if they are not the same, so that invoice data can be merged directly into ledgers.

Fleet Management

The area of fleet management is one that is specific to transport operators and concerns the requirement of operators to maintain information about vehicles, their use, performance, maintenance and costs. Spooner (1988) lists a number of additional features such as workshop control, warranty control and tachograph analysis which are desirable in such a system He states that in 1988 about 50 companies were offering such packages, and lists some 11 that have been available for some time

While this is an area which is clearly required by all transport operators, such packages are usually targeted at operators of large fleets of vehicles, and they may be inappropriate or too expensive for smaller CT operators. Nonetheless, there is no doubt that they provide useful facilities for monitoring vehicle costs and performance and for administering the fleet.

Booking or Despatching

This is one of the areas which is most critical for CT operators. Most bookings for services are taken over the telephone, and there is a requirement from operators for a system which speeds up the process of handling bookings, and which reduces the amount of paperwork The booking task concerns the entry of information from the passenger or hirer onto a form, while the despatching task involves the decision as to whether that request for transport can be met and, if so, by which vehicle or service. For group hire and furniture transport, this may be a simple decision as to whether a vehicle is available at the time and on the date requested. For passenger services, the decision may be a more complicated one, involving several vehicles all of which could possibly carry the passenger, and where the problem is to minimise the cost to the operator and to other passengers, in terms of increased travel times, and where the despatcher may be involved in negotiations with the would-be passenger over pick-up and drop-off times. Approaches to this task and the logistics are covered by Potter and Hagyard (1986). Most paper-based systems involve the entry of information onto forms and the entry of some minimal information about each trip onto a separate sheet of paper or wall-board representing vehicle shifts. A considerable amount of booking clerks' or despatchers' time is taken up with making fair copies of the shift records or schedules after all the bookings have been taken, possibly attempting to improve on the schedules at the same time.

Meadows (1987) covers the arguments for the computerisation of dial-a-ride bookings in some detail, specifically to improve manual systems dependent on index cards and paper, to speed up the despatch process and to make data on trips available for analysis as part of a process of service development, forward planning and marketing The need for a computerised system is probably greater in dial-a-ride and dial-a-bus systems where a high volume of bookings must be taken, often in a short period of time. However, computerising group hire and other services should provide benefits to the operator in terms of automatic checking of vehicle availability, consistency of data across all services, and shortfall and other monitoring. There is no off the shelf software available to perform these functions.

Routeing

For group hire services and for dial-a-bus services, the problem of selecting the shortest or cheapest route between a number of points is a common one, although it is not usually perceived as a routeing problem. In group hire services, it is often a requirement of a hirer that a number of passengers are to be picked up and delivered to a common destination This is often the case for transport to day centres and luncheon clubs for elderly or disabled passengers. For dial-a-bus services, a number of passengers in a particular area may have to be picked up and delivered to a common destination such as a shopping centre. In both cases, there are advantages to optimising the route taken to reduce the distance travelled by the vehicle and thus to reduce costs. Bodin et al. (1983) provide the most comprehensive survey of such problems, and of algorithms for their solution

Most packages which implement such algorithms operate over wider areas than the operational areas of CT operators. There are a number of such packages available which address the problem in the context of freight transport over large distances between cities. In the UK the package Autoroute is now available as an off the shelf package which provides routeing capabilities. However, it is not really suitable for transport on a local, intra-urban basis.

Scheduling

While the problem of routeing concerns the organisation of passenger pick-ups or drop-offs in space, scheduling introduces the dimension of time, usually in the form of constraints: either those imposed by passengers' requirements to reach their destinations by certain times, or by common sense requirements to minimise travel time for frail elderly or disabled passengers.

For dial-a-ride operators, the scheduling task is a complex one and one which is often avoided because of the effort involved in re-scheduling passenger trips that have been accepted and despatched to specific vehicle shifts. Most dial-a-ride operators in the UK operate a first come first served booking system which can result in inefficiencies, as passengers are allocated space on vehicles according to their ability to ring up early in the booking period or to their persistency in re-dialling when lines are engaged. A number of American systems are designed on the basis of taking all bookings and then scheduling them later. The work of Alfa (1986) and Kikuchi (1987) is typical of this approach. It does, however, require the operator to ring passengers back to confirm bookings or to offer alternatives, and in this sense may be dependent on the North American telephone system, on which all local calls are toll-free.

While there are a number of algorithms for dial-a-ride scheduling which have been surveyed by Bodin et al. (1983), and which have been implemented in some American systems as research projects, there were no commercially available packages to provide this function at the start of this research.

Performance Monitoring

The issue of performance monitoring follows from the availability of data in a computerised system, and the requirement of operators to be able to analyse that data for the purpose of reporting on past performance, possibly to justify funding from public or other sources, in order to monitor performance and costs with a view to increasing efficiency, and also for the purpose of forward planning and service development.

Once the data associated with the other functions outlined above has been entered into a computer system and stored there, it should be accessible to users in a way which allows it to be sorted, selected and summarised for this purpose. Evans (1987) outlined the requirements for a Management Information System The way in which this is done is dependent on the choice of software to perform these other functions. The availability of data in the database held by users of Fletcher's D-A-R has led to some research into patterns of use in Manchester, Haringey and Nottingham among others (Campbell, 1987; Sutton et al., 1987; Sutton, 1989a). While this may be of use in analysing usage patterns, identifying shortfalls, and planning services, it can also lead to attempts to analyse productivity in purely quantitative terms. Arguments for quantitative and qualitative evaluation can be found in papers by Smythe (1988

& 1989) and by Sutton (1989b). The issue concerns the identification of the proper focus of evaluation, and of the balance between efficiency and effectiveness in an overall analysis of a service. It is important to recognise that computerised data, while it may help in evaluating efficiency, is unlikely to be of help in evaluating social effectiveness.

The Problems for CT Operators

This research aims to tackle four areas which pose problems for the computerisation of community transport administration and operations. Each of these will be covered in more detail in Chapters 3 and 4. However, a summary of each follows

Integrated Data Structure for all Services

The software packages for CT services available in the UK at the start of this research, in 1986, only catered for one type of service in each case. For many operators, this was a serious consideration in determining whether they should invest in a computer system including ready written software, or in the development of software to meet their own needs The aim of the research is to provide a package which provides for the range of services offered by CT operators. It is proposed that this software package should be both modular and integrated: modular, in that each element of the system catering for a specific service can stand alone, independent of other modules; integrated, in that each element of the system will be compatible with every other. In order to achieve this, a data structure is required which encompasses all the services and the data involved in booking and despatching passenger and vehicle trip requests The designs of the databases which underlie software catering for single services are not adequate for software catering for multiple services, and the fundamental design work is necessary to establish a data structure which can support an integrated package

Brokerage

An area not outlined specifically above, but which covers aspects of booking, accounts and invoicing, and where there is potential for development of computerised approaches which improve efficiency is that of vehicle brokerage. A number of CT group hire operators offer a brokerage service to their members and to owners of other vehicles in the voluntary sector. As such, group hire can be seen as an operational subset of a vehicle brokerage service in which all the vehicles belong to a single operator. Vehicle brokerage poses additional problems in accounting for income generated by bookings taken on behalf of another operator, and in reconciling different methods of charging for vehicle use which may be applied by the broker and the vehicle operator.

A further technical challenge is posed by the current lack of any methods for automating the selection of vehicles to match a hirer's requirement for travel. The development of computerised techniques to handle this becomes a possibility when data concerning vehicle availability and facilities are held in a computer system.

Routeing

There are a number of established techniques in the field of operations research to implement computerised vehicle routeing (Bodin et al , 1983). Most of these have been developed either as academic exercises or to provide solutions to large scale routeing problems in transportation or production management, for example to minimise the distance traveled by a drill head drilling holes in a printed circuit board. Despite being computationally intensive, most such solutions provide only near-optimal solutions. Given this constraint, it may be appropriate to employ other techniques, in particular the space-filling curve algorithm, mentioned above in the context of TRANSIT, and covered in detail in Chapter 4, which only requires sorting of numeric values associated with spatial coordinates in order to produce a near-optimal route.

The research, therefore, aims to provide a simple solution to the problem of vehicle routeing on the small scale, and to implement and test this as part of the software package.

Scheduling

In a similar way, there are a number of algorithms which have been developed, particularly in North America, to solve the problem of scheduling dial-a-ride trips These are predominantly batch-oriented, taking into account the North American approach to taking bookings, scheduling them, and ringing passengers back to confirm or cancel. They are not dynamic or interactive.

This research will examine existing manual scheduling techniques in the UK and, as a first step to their computerisation, develop approaches which allow the computer to be used as a tool by despatchers or schedulers firstly to improve their productivity and secondly to improve vehicle productivity by matching passenger trips more appropriately to existing schedules.

It is also intended to identify areas for development of scheduling tools and techniques which could form the basis for further research, using the structure of the software package developed here.

These four areas form the basis for subsequent chapters. However, before proceeding to develop these ideas, Chapter 2 outlines some of the approaches to the research and the techniques and software tools used

Chapter 2

Methods

General Points

As a piece of software development, the research does not have a single methodology as might a piece of research based on interviews or questionnaires. However, the approach reflects a number of research traditions within the overall field of qualitative research.

Firstly, the author's previous research experience has been of action research in community work and small group work This approach has a lot in common with the idea of a pilot project in terms of its tentative nature, and, as stated in the introduction, one of the aims of the research was to test out the software in pilot projects with collaborating organisations. This tentative nature is one of the characteristics of action research identified by Clark (1972) in contrast with summative evaluation This is shown in Table 2 1.

Summative Evaluation	Action Research
1 Requires a clear and constant purpose	1 Tentative, non-committal and adaptive
2 Logical development of steps	2 Focused on the next stage
3 Perceives the present situation in the context of the final outcome	3 Evolves the future out of emerging opportunities
4 Is not able to interpret thepresent until it knows the answer to its ultimate questions	4 Has to interpret the present as a basis for asking questions
5 Focuses upon a limited range of factors	5 Attempts to comprehend a wide range of factors in a dynamic relationship

Table 2.1 *Summative evaluation research and action research: key differences (Clark, 1972)*

The characteristics of action research listed in the table are very similar to those of the rapid prototyping approach to software development which has become possible with the development of software tools such as applications generators and fourth generation languages, such as System Builder (described below), which make it possible to develop an application in outline much more quickly than used to be the case with earlier software tools. In the systems design context, the contrast is with the systems life-cycle approach to software development, which provides a rigid framework of development stages through which an application must pass during the course of its development. The rapid prototyping approach, on the other hand, is both iterative and more interactive in its nature.

Secondly, the research is predominantly qualitative in nature. Although a number of CT operators were visited, their documents collected and analysed, their staff interviewed and their working practices studied, the extension of the research to a number of operators was not for statistical purposes It was rather in order to gather more and different examples of the ways in which information is used by different operators. The approach fits closely the style described by Fielding and Fielding (1986) in relation to qualitative research methods, and shown in Table 2.2.

Preferred Data	Qualitative	Quantitative
Most common analytic technique	Analytic induction (e g grounded theory)	Hypothetico-deduction
Logic of generalization	Generalization by examining (many cases of) data to determine axiom that fits all cases	Generalization by winnowing hypotheses in testing against data to see how many cases it explains

Table 2.2 *The inter-relation of data (Fielding and Fielding, 1986)*

This generalisation was carried out in a synoptic and synthetic way, as an attempt to develop a data model that could be used in a wide range of CT operations. In this sense, the work is relativist rather than absolutist in its approach to the data model.

Thirdly, this work is not concerned with the replication and testing of an existing model, but with the development of a model which can be used as the basis for the design of a computerised CT system directly from the manual administrative and business practices of CT operators None of the other software packages discussed either in the survey in the previous chapter or in more detail in the next chapter is concerned with the development of a model which will form the basis of subsequent system design, but only with the solution of immediate problems. The development of a data model for CT operations can form the basis for further research in this field, for example in performance monitoring, in simulation as a means of planning new services, in automating the routeing or scheduling process, and as the underpinning for a knowledge base to allow the implementation of expert system approaches to the solution of operational problems.

Despite the lack of an overall methodology, there are a number of areas of the research which reflect either a particular approach to or a particular tool used for the analysis, the design or the development of the computer package. The rest of this chapter explains these issues, which are best divided into three types:

- Collaboration with CT Operators;
- Analysis and Design Techniques; and
- Software Tools

Each of these will be covered in turn.

Collaboration with CT Operators

Before describing the use of collaboration, it is necessary to explain why this approach was adopted.

Firstly, community transport in the UK, as has been noted above, is largely based in the voluntary sector. Moreover, it has its roots in community development and self-help, and many CT projects have arisen from the work of paid or unpaid community workers with community organisations As such it is committed to participative management; most CT operators are run by management committees which include representatives of users of the services, whether individual passengers in dial-a-ride or dial-a-bus services, or representatives of member groups in group hire services. Given, the author's own background and experience in community work, it was felt that it would be inappropriate to carry out the research in a way that did not involve, directly as participants, those people and organisations that would be affected by the outcome, as users of the software.

Secondly, while the development of software of this scale and nature without the involvement of end-users would be impossible, and work with users is an inevitable element of any systems analysis process, the level of participative involvement of a number of users, perhaps reflects the commitment to participation described above

Thirdly, the IT in CT Project had in its previous research already adopted an approach which involved users of computers and computer software from CT in the work of the project.

Fourthly, the approach of the research was, as outlined above, synthetic and concerned with the derivation of a data model from as wide a range of operating practices as possible. The collaboration of a number of CT operators was an important element in providing a diversity of practices on which to base the analytic process.

The development of the software involved CT operators in two formal collaboration processes, and these were supplemented with informal feedback. The two formal approaches were:

- a User Group; and
- Collaborative Projects

User Group

The User Group had its origins in the previous work of the Information Technology in Community Transport (IT in CT) Project carried out by Dr Sutton, and referred to briefly in the Foreword. The User Group originally consisted of a group of people with an interest in the use of computers in CT. They were not necessarily themselves users of software for CT, although most of them were connected with organisations which used Fletcher's D-A-R, and their predominant interest was in the use of the data derived from the use of computers in CT in order to monitor and plan services.

When this research started, the User Group took some responsibility for representing the ınterests of the CT sector in the development of the software, although it was primarily responsible for Dr Sutton's continuing research. The User Group had done some prelimınary analysis of the file and data requirements for the software to be developed, although this lacked analytıcal rigour and was revised substantially.

The main functıon of the User Group was, in conjunction with the analytıcal methods described in the next section, to resolve the differences in operating practices found in dıfferent CT operators into a single model which could be converted into a specification for a computer system. This was an iterative process which is shown in dıagrammatıc form in Fıgure 2.1.

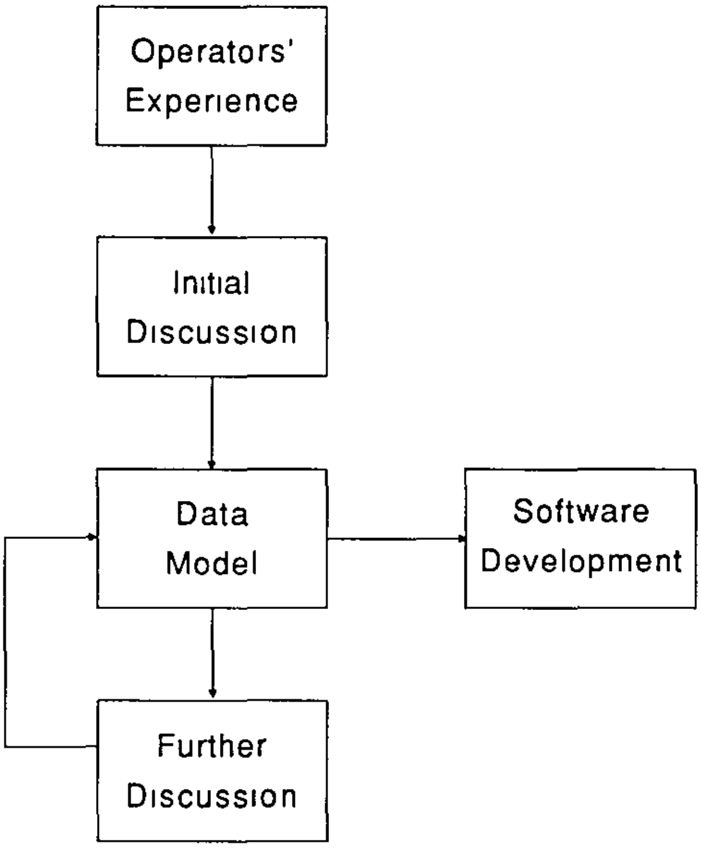


Figure 2.1 *The iterative process of workıng ın the User Group*

Each of the members of the User Group was involved as an operator ın or in research into different CT organısations across the UK Their experience of different operational styles and practices was used as the basis for the initial discussions. The way in which this experience was used to start the process of defining the data model is shown in Figure 2.2., and this process is discussed in more detail below. The results of the initial discussions were used to develop a first pass logical data model, using the tools of entity-relationship analysis (described in the section on design tools below), and this was presented back to the User Group, using charts and diagrams. Further discussion was used to refine the model. For example, the assertion that 'a passenger pays for a dial-a-ride trip' would prompt someone to say that it could be an organisation such as a social services department that paid for a trip, and the model would be amended accordingly if this point was accepted by the User Group. The data model produced by this process then provided the basis for the development of the software.

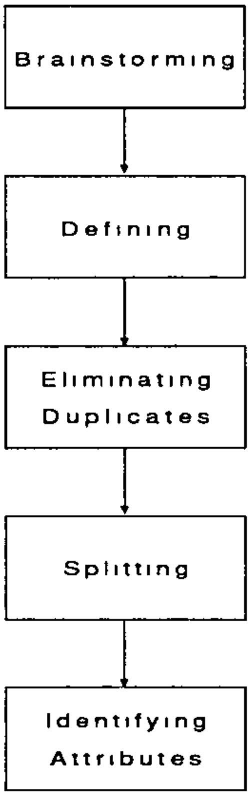


Figure 2 2 *Analysing files in the User Group*

The way in which the discussion process took place in the User Group, referred to above and illustrated in Figure 2.2, reflected both the need to develop a model which encapsulated different working practices in different CT organisations, and the author's community work experience and approach to work with the group. This process is explained as follows.

The first stage of the process was to identify an area of CT operation, possibly a whole service such as dial-a-ride, and to brainstorm on the subject of what objects or entities were involved in the operation, which might be the subjects of files in the computer system. These would include physical objects and people such as vehicles or drivers as well as abstract objects such as shifts.

The second stage of the process was to define each of these, and to identify the approximate quantities involved. For example, it might be agreed that a vehicle was 'any vehicle, whether specially adapted or not, used to provide transport as part of a dial-a-ride service'. This led naturally to the next two stages, when it was discovered that some objects which had been given different names by different operators were in fact the same, and the duplicates could be eliminated, or more often when it was discovered that two operators referred to different entities by the same name, and these were split. For example, different operators used the term 'escort' to refer to (1) a paid or unpaid driver's assistant who travelled on a vehicle to assist the driver and the passengers, (2) a paid or unpaid person who met a passenger at his or her destination and helped him or her, often by pushing a wheelchair, or (3) a passenger's own personal escort who might be a relation or friend who travelled with the passenger. Each of these would then be given a separate name and uniquely defined The elimination of duplicates also allowed for the elimination of entities which were felt by the User Group to be outside the scope of the project, or not part of a computerised system.

The final stage involved identifying what information should be stored about each of the entities which was to be part of the computerised system, for example, passenger name, address, date of birth. While much of this was straightforward, some was more contentious, for example the issue of whether or not to maintain information about a passenger's ethnic origin in the computer system, and if it was to be maintained, how it was to be maintained. The involvement of a group of operators was expected to assist in resolving conflicting requirements.

After the involvement of the User Group in the initial design phase, it continued to play a role as the software was developed. The rapid prototyping which was possible with the use of the System Builder applications generator meant that data models could very quickly be turned into files, screen designs and reports and output forms, and that these could be demonstrated to the User Group. This led to feedback on the designs and further refinement of these designs.

Collaborative Projects

The Collaborative Projects were represented in the User Group. It was initially intended for the first phase of the research, concerned with the development of software for group hire services, that three operators would be involved as Collaborative Projects. These three groups had been selected as a result of the work of the User Group in the previous research of the IT in CT Project. They were Birmingham Community Transport, Enfield Community Transport and Nottingham Community Transport. However, none of these three operators were in a position to collaborate fully, as they did not have the computer hardware necessary to be involved. Both Birmingham and Enfield CTs were applying for grants to cover hardware costs, and Nottingham CT had a computer, an IBM PC/XT used to run Fletcher's D-A-R which was already overloaded with data and barely able to keep up with the growth of Nottingham CT's dial-a-ride operation.

Two other operators approached the project and were accepted as Collaborative Projects These were Camden Community Transport which provides a group hire service and a social car scheme, and Barnsley Dial-a-Ride which provides a dial-a-ride service and a social car scheme and which had plans to provide a dial-a-bus service Both organisations had the necessary funding to go ahead with the purchase of computer hardware at the appropriate times, and thus became involved as full Collaborative Projects.

Other collaboration

Within the CT sector, there were effectively four levels of collaboration with the research.

1. The full Collaborative Projects which took on the nature of Pilot Projects

2. The initial Collaborative Projects and members of the User Group which provided material for some of the research and analysis.

3. The first customers who bought the software, with whom there was considerable involvement on the part of the research project.

4. Many other CT operators who came to Loughborough to look at the software, or who saw it at demonstrations and trade shows and who provided valuable informal feedback.

From the point of view of the research, it was the first three levels which provided the bulk of material, and particularly the two pilot projects where much of the detailed analysis was carried out. In contrast to the work with the User Group, the work with the Collaborative Projects used the more usual tools of systems analysis, such as interviewing, observation and the collection and examination of existing forms and documents. These were used with the more formal tools of procedure charting and data flow diagramming to document existing procedures and to design computerised procedures.

An important consideration in this was the User Group's concern that in designing a computer system which was appropriate to many different styles of CT operation, existing manual practices should not be incorporated into the design without question, but that the computer system should attempt to impose standards by incorporating good practice from a number of operators

Analysis and Design Techniques

Systems analysis has always been concerned with the use of documentation tools as aids to the process of analysis and design of computer systems. During the 1970s and 1980s, a number of more formal, structured approaches to systems analysis have been developed and, in most cases, marketed as solutions to the problems of designing and documenting large systems by their authors. Jackson Structured Programming (JSP) is an example of such a system, marketed in the form of training courses for data processing professionals. During the 1980s, the concern with formal structured approaches has spread, and in the UK has been most apparent in the promotion of a methodology known as Structured Systems Analysis and Design Method (SSADM). The origin of SSADM lies with Learmouth and Burchett Management Systems (LBMS) who developed it in conjunction with the UK

Government's Central Computer and Telecommunications Agency (CCTA). It is now a requirement for most software development projects for central and local government departments and agencies.

SSADM consists of a series of stages in the development of software and a number of analysis and design tools which are to be used as part of the documentation of each stage (Downs, Clare and Coe, 1988) The justification for formal methodologies such as SSADM lies in the scale of many software projects, and the need for an approach which allows many individuals in teams to tackle elements of the project. As such, it is prescriptive, laying down precisely what is to be done in each stage, and reductionist, breaking problems down into small achievable tasks. It is, however designed to be tailored to the project in hand, allowing some choice of techniques while not allowing important stages to be bypassed.

SSADM is designed for large projects involving large numbers of people, and neither it nor any other formal, proprietary methodology was adopted in the research However, specific analysis and design techniques were chosen for the purpose of documenting the process of deriving the data model and developing the software package These tools and the reasons for their choice are described below

Three formal systems analysis documentation tools were used in the analysis and design of the system. These were·

- Entity-relationship diagrams,
- Procedure charts; and
- Data flow diagrams

These techniques were used for a number of purposes:

- to document the research and the analysis,
- to provide graphical feedback to the User Group and Collaborative Projects;
- to highlight graphically differences in operating practices between different CT operators; and
- to reconcile different practices into a single operating model.

Other techniques such as decision tables and program flowchartıng were also used, but more as programmer's tools than as design tools. The three main techniques are described below.

Entity-relationship diagrams

Entıty-relationshıp (E-R) diagrams are used to produce a logical model of the files ınvolved in the process being analysed. This tool was used largely in conjunctıon with the work carried out with the User Group in identifying the objects or entities involved ın CT operatıons. E-R diagrams were first proposed by Chen (1976) and are widely used and described in many systems analysis and data processing texts (Bingham, 1983), and many authors have proposed theır own refinements of the technique (Avison, 1984, Curtıce and Jones, 1982; Howe, 1989). The technique used in the research and descrıbed here ıs essentially that descrıbed by Veryard (1984), and referred to by Downs, Clare and Coe (1988) in theır text on SSADM as Logıcal Data Structurıng Technique (LDST). A straıghtforward version of the technique wıthout addıtıonal notatıon was chosen so that diagrams would be easıly understood by members of the User Group.

Downs, Clare and Coe gıve three objectıves for the use of LDST.

- To document the data requırements of the system under investigation.
- To diagrammatically represent the relationships between data.
- To aid communıcatıon

The constructıon of an E-R chart follows the stage of identifying the entitıes, whıch ın the case of thıs research was carried out in conjunction wıth the User Group After the entıtıes have been lısted, relatıonshıps between them are identified, and may be charted in a grıd. The grıd can then be used to produce the LDS or E-R diagram. In the diagram entıtıes are represented by rectangular boxes, lines between entıtıes represent relationships, relatıonshıps can be one-to-one, one-to-many or many-to-many, and the many end of a line is marked wıth a crow's foot; many to many relationships are normally converted into two one-to-many relatıonshıps, through the introduction of a new entıty to represent the link; optional relationships are shown by an 'o' on the lıne. Fıgure 2.3 shows the elements used ın an E-R chart.

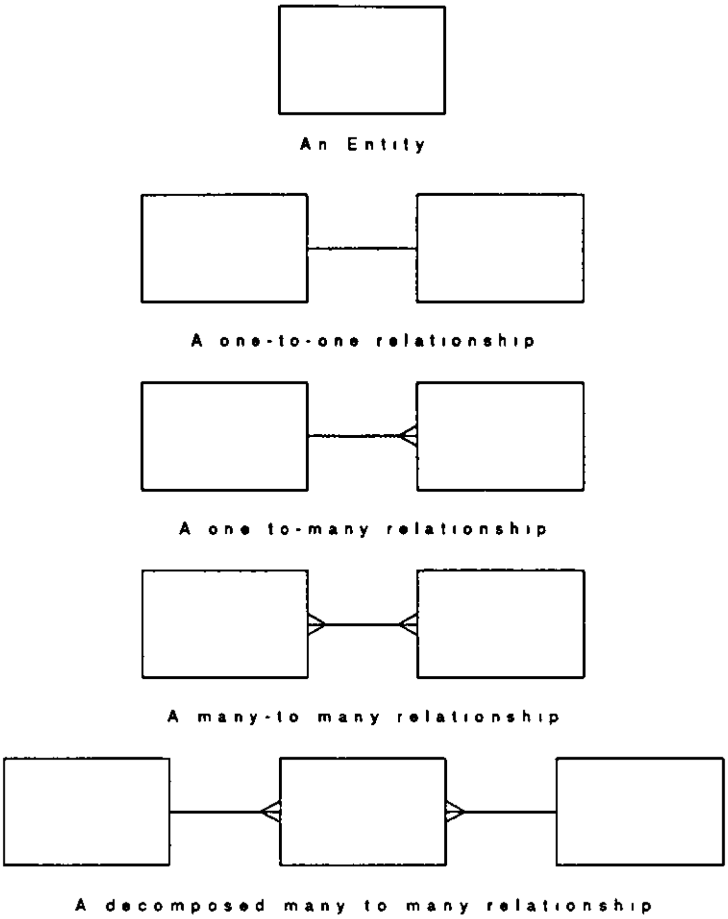

An Entity
A one-to-one relationship
A one to-many relationship
A many-to many relationship
A decomposed many to many relationship

Figure 2.3 *Elements of an entity-relationship (E-R) chart*

As an example, in a vehicle brokerage scheme, a member group may own no vehicles, or it may own one or more vehicles However, a vehicle can only belong to one group. The relationship between member groups and vehicles is an optional one-to-many relationship. A member group may have any number of volunteer drivers who drive for that group, and each driver may drive for several groups. The relationship between member groups and drivers is a many-to-many and is represented by the introduction of a new entity ‘member group driver’. These two relations are shown in Figure 2 4, which is part of a much larger chart for group hire and vehicle brokerage.

Each of the boxes in the E-R chart can be seen as representing a logical file in the system. Further techniques such as normalisation of the data model (Codd, 1970) can be applied, in order to eliminate duplicate data. The logical data model produced by E-R charting forms the basis for the design of a physical file structure within the computer system.

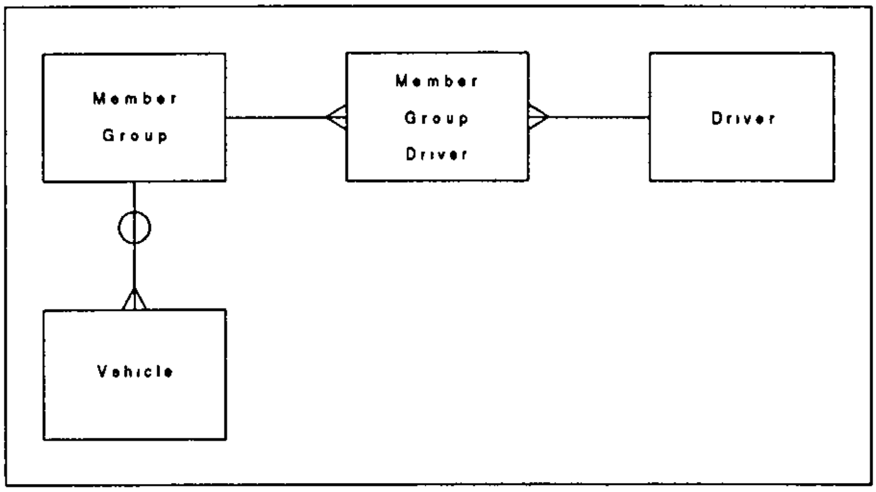


Figure 2 4 *Part of an E-R Chart*

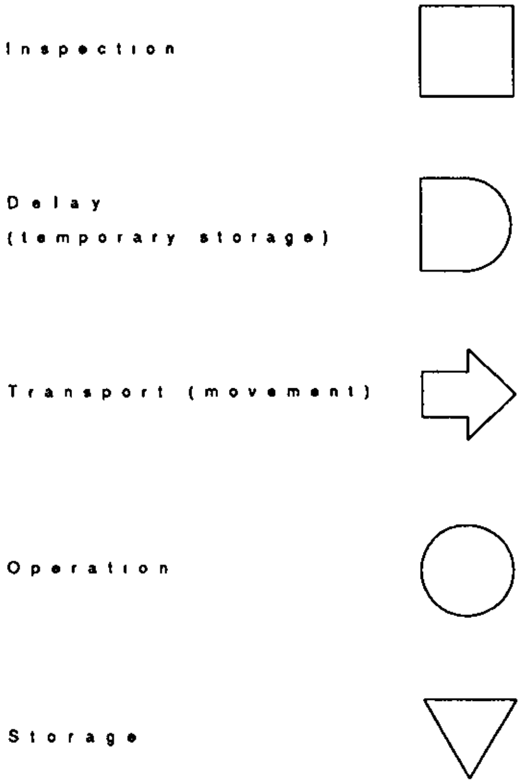


Figure 2.5 *Symbols used in procedure charts*

Procedure charts

Procedure charts provide a means of recording and descnbing the flow of documents and ınfonnation in exıstıng systems, and of outlınıng the manual procedures to be used with a computer system. Bingham (1983) suggests that they provide a useful way of ıdentıfyıng and analysıng bottlenecks. The most widely used procedure chartıng technıque ıs based on the use of the five symbols promulgated by the American Society of Mechanical Engineers (ASME). These are shown above in Figure 2 5.

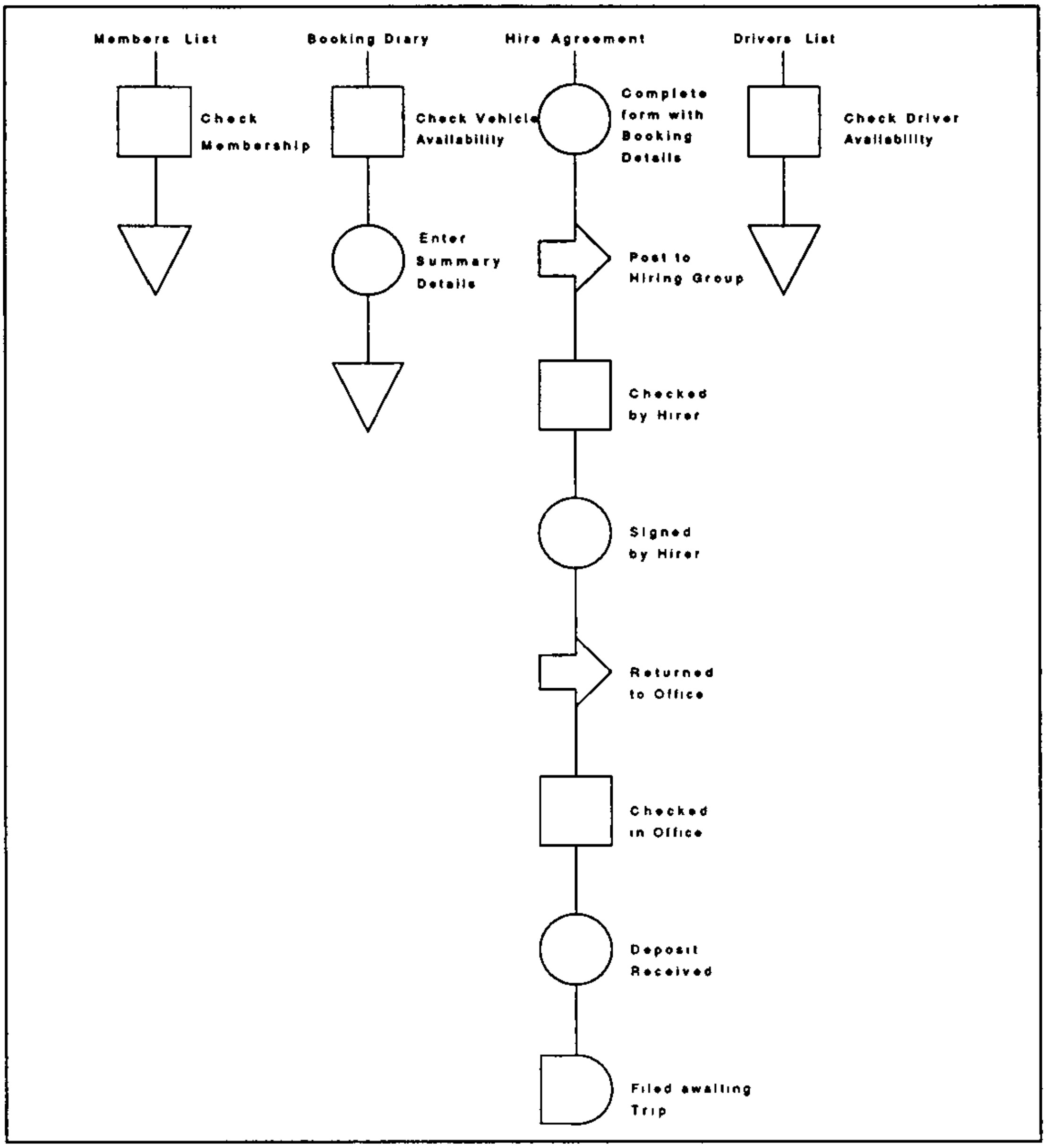


Figure 2.6 *An example of part of a procedure chart*

The five symbols represent possible manual procedures applied to documents or other information in a system. A square represents an act of inspecting or checking a document; a circle represents some other operation, such as entering information onto the document; a 'D' shape represents temporary storage, usually filing pending some other action or batch process; an arrow represents a movement of the document, possible from one department to another, and a triangle represents permanent storage. Figure 2 6 shows an example of a procedure chart for taking a group hire booking in a community transport operation.

Data flow diagrams

Data flow diagrams (DFDs) are used to analyse and illustrate flows of data in an information system. They can be used to document existing manual systems as well as modellıng computerised systems, and are concerned primarily with the functions withın the system and the data used by those functıons. Downs, Clare and Coe give five objectıves for the use of DFDs

- To document the boundaries of the system.
- To show the movement of data between the system and its environment.
- To provide a hierarchical functional breakdown of the system.
- To document the intrasystem information flows.
- To aıd communication.

DFDs are more complıcated than E-R charts and five symbols are used to construct the diagrams These are shown in Figure 2.7.

Data flows are shown by arrows and represent the flow of data from source to destination. Physıcal flows are shown by outlined arrows and represent the flow of physical goods or servıces around the system. Functions or processes are shown by numbered and labelled boxes and represent the transformation of data flows in the system Data stores are shown as narrow open-ended rectangles, whıch are numbered and labelled and whıch represent stores of data which could be files or card indexes ın a manual system or computer files ın a computerised system. Finally external entıtıes are shown as ovals and represent entıties such as customers who are the destinations or sources of data flows or physıcal flows. Fıgure 2.8 shows an example of a DFD. DFDs are hierarchical: a system can be represented as a single context or overvıew diagram; each process wıthin a DFD can be broken down into sub-processes, until the level of detail requıred is reached, and a specification for each process can be produced either as structured englısh or using a technıque such as decision tables or decısıon trees.

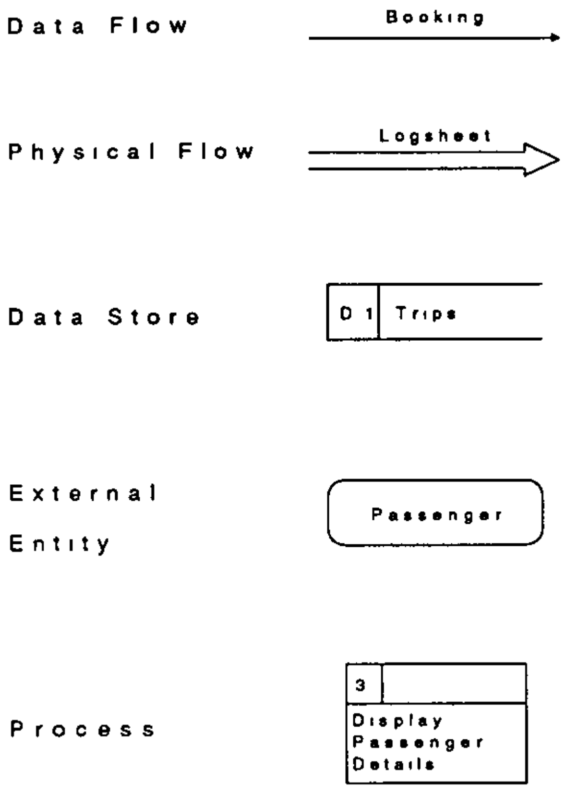


Figure 2.7 *Symbols used in data flow diagrams*

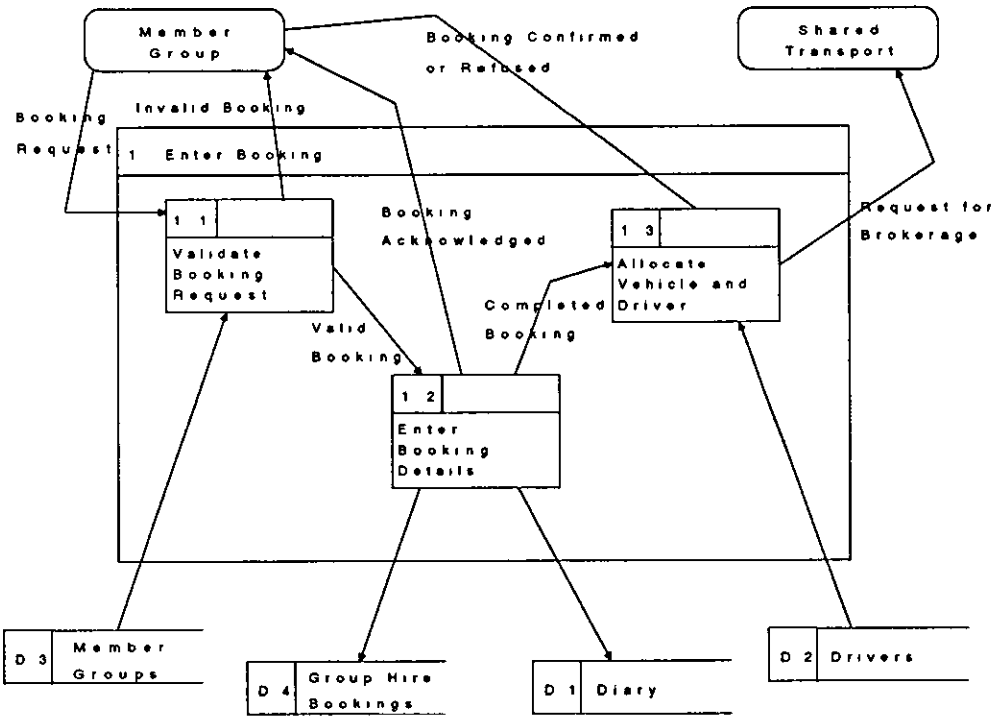


Figure 2 8 An example of a data flow diagram

Each of these three analysis techniques was used in the research to analyse the working of CT operations, to present the analysis to users, to help in the design of the computer software and to document the design.

In addition, a further charting technique has been used in this thesis in Chapter 3 in order to provide a summary overview of some computerised systems. Figure 2.9 shows the symbols used and their meanings.

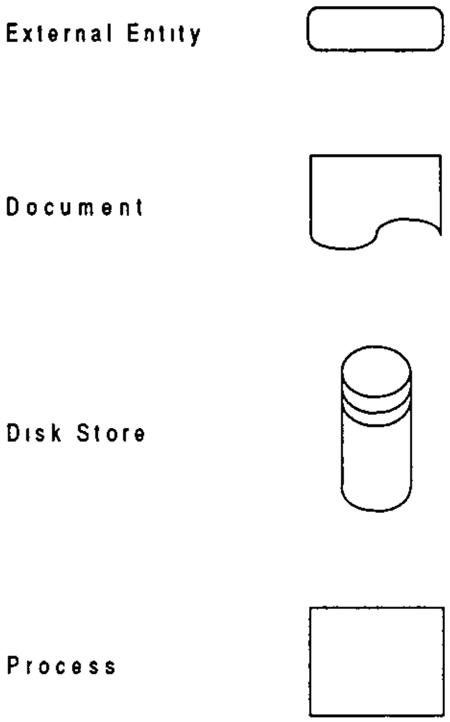


Figure 2.9 *Symbols used in comparing packages in Chapter 3*

Software Tools

Two software tools were used to develop the software as part of the research:

- the PICK operating system; and
- the System Builder applications generator.

Each of these will be described, and the reasons for their choice outlined

PICK operating system

An operating system is a software package which runs on a computer and which provides system level functions to do with the housekeeping of the computer and its associated peripherals, for example organising the storage of data on disks or tape or handling input from keyboards and output to screens. Some operating systems are proprietary and only available on certain manufacturers' computers, for example VMS on VAX computers from the Digital Equipment Corporation (DEC), or VME on ICL mainframes. Other operating systems are available to run on computers from a number of manufacturers, for example CP/M on computers using 8-bit Intel 8080 or Zilog Z80 microprocessors, MSDOS or PCDOS on IBM PC compatible computers using Intel's 16-bit family of microprocessors (8088, 8086, 80286, 80386), or UNIX which runs on a range of microcomputers and mini-computers.

PICK belongs to the latter family of operating systems. Like UNIX it will run on a wide range of makes and sizes of computers, from PCs through mini-computers to mainframes. Unlike CP/M and MSDOS it is not restricted to any particular processor. Unlike almost any other operating system, PICK contains a number of features which make it particularly appropriate for the development of database oriented systems. These have been covered by a number of authors (Cook and Brandon, 1984a & b, Taylor, 1985; Bate, 1986, Bull, 1986; Bourdon, 1987). They can be summarised as follows

- The operating system is optimised for handling data.
- A data dictionary is a feature of the operating system.
- An enquiry language known as 'ACCESS', 'ENGLISH' or 'RECALL' is provided as part of the operating system, allowing the user to enquire of the database and produce detailed reports.
- The programming language PICK DATA/BASIC is closely integrated to the database.
- Records and fields within records are variable length, allowing efficient use of disk space
- A comprehensive job control language known as 'PROC' is part of the operating system.

From the point of view of the research, there were a number of other reasons for choosing PICK as the operatıng system for development of the software.

- PICK is a multi-user operating system, allowing many users to share the resources of a single machine, using terminals.
- PICK will run on a wide range of types and sizes of machines.
- Programs written in PICK DATA/BASIC are portable across the range of computers whıch support PICK.
- On an IBM compatible PC, it is possible to run both PICK and MSDOS, although not concurrently.
- Fletcher's D-A-R, whıch at the start of the research was the software package for CT most wıdely used, is also wrıtten usıng PICK, and experience has shown that it is easy to use.
- The ACCESS enquıry language provides a valuable facility for analysıng the data ın the database.

Since the start of the research, there has been a further development in relatıon to the PICK operating system. A number of lıcensees of PICK have started to provide versıons of PICK which wıll run eıther as a process under UNIX or on a machıne concurrently with UNIX. For larger systems, thıs gıves the opportunity to run software under eıther operating system and to pass data between the two. Implementatıons of PICK are created for particular computers by lıcensees who are licensed to do this by PICK Systems Inc. In addition, there are two related operatıng systems whıch are very similar, but which are not PICK. These are PRIME Information whıch runs on mini-computers manufactured by the PRIME Corporatıon, and REALITY which runs on mini-computers manufactured by McDonnell Douglas Information Systems There is also a database management system written to run as an MSDOS applıcation, which is very sımılar to PICK. Thıs is Revelation, from Revelatıon Technologıes Inc.

While the PICK operatıng system has been optimised for database handling, software to run under PICK can either be written in the PICK DATA/BASIC programming language or wrıtten usıng one of a number of applıcatıons generators, whıch are sometimes called Fourth Generation Languages (4GLs).

Language generations

Computer languages are considered to fall into 'generations'. The first generation is the machine code, which forms the underlying language of all computers, and in which the earliest computers were programmed, often by painstakingly entering the binary machine language through a set of switches directly into memory.

The second generation is assembly language in which machine code instructions are given mnemonic identifiers, such as ADD or JMP, and in which machine addresses can be given meaningful names. Assembly language allows the writing of programmes which humans can begin to make sense of, and which are easier to enter, although there may still be an effective one-to-one correspondence between the mnemonics and the machine language

The third generation comprises the bulk of computer programming languages used today. FORTRAN, COBOL and ALGOL are examples of the earliest such languages. In these languages, also known as high level languages, single instructions represent complex actions which may be translated into tens or hundreds of individual machine language operations

The fourth generation is not clearly defined, however, one view is that such languages are concerned with stating 'what' is required rather than 'how' it is to be done In effect, an additional level is introduced between the user or programmer and the hardware Whereas in a conventional third generation language, it is the responsibility of the programmer to detail quite precisely how a program is to perform the actions required of it In a fourth generation language it may only be necessary to state 'what' is required, and the program deals with the 'how' either in a predefined way, or in different ways depending on the context Examples of fourth generation languages are enquiry languages which allow the user to extract data from a database by simply stating what is required·

'SORT CUSTOMERS BY BALANCE WITH BALANCE > "2000 00" DISPLAYING CUSTOMER.NAME CUSTOMER.PHONE CREDIT.LIMIT TOTAL BALANCE'

The ACCESS enquiry language in PICK is considered by many to be a 4GL. Probably one of the best known enquiry languages is SQL, which is used in IBM's mainframe DB2 language, and which is now an American National Standards Institute (ANSI) and International Standards Organisation (ISO) standard (Groff and Weinberg, 1990).

Applications generators

Another family of programming tools which some consider to be 4GLs are applications generators. These are packages which allow the programmer to write applications programs by defining the 'what' rather than the 'how'. For example, data entry screens can be designed using a screen-painter, in which fields are placed on the screen possibly by using a moving cursor to choose the screen position and by defining the fields in a data dictionary. The applications generator may then run that data entry screen program using a file of information, a parameter file, and the data dictionary, or it may use the parameters and the data dictionary to 'write' a program using a set of standard routines. In some cases, such programs are completely stand-alone, and can be run without any software from the applications generator present on the computer, in other cases, they require at least a run-time module of the applications generator present to provide support subroutines.

For the software developer, there are disadvantages and advantages to the use of a 4GL or applications generator, compared to the use of a traditional third generation programming language.

- Applications generators which use parameter files can be slower than compiled programs, as they have to interpret the parameters as they are running
- Applications generators can introduce overheads in terms of calling the standard subroutines which they use to perform certain functions.
- Applications generators which generate compilable programs can produce programs which contain code which will never be used.
- Many applications generators are inflexible in the facilities they provide, for example in validating data typed in by the user.
- Unless they provide particular points at which it can be linked in, applications generators may not allow the programmer to include sections of program written in a third generation language where necessary.

- Applications generators do produce programs which are consistent in the way they present to the user, for example in screen layout or the provision of help.
- Most applications generators will produce printed documentation describing the package developed.
- Packages developed using applications generators are usually easier to maintain than traditional programs
- Because the programmer is not concerned with details of how the program works, it is usually quicker to develop packages with an applications generator.

For this research project, the fixed timescale and the fact that only one person was available to carry out the development meant that an applications generator with its documentation capabilities and faster development times was the preferred choice, although it was also considered important that the applications generator chosen must allow the integration of PICK DATA/BASIC code with the generated programs for maximum flexibility. A list of applications generators for PICK which were available at the start of the research is shown in Table 2.3.

		Machine	UKPUA	
Product	**Supplier**	**Specific**	**Seminar**	**Type**
ALL	McDonnell Douglas	Y	Y	P
CREATOR	Logical Choice	N	Y	C
CUMULUS	Velvet International	N	N	P
Custom-Writer	Business Logique	N	N	PC
Libra	Simdell	N	Y	PC
Paradyme	Computerstyle	Y	Y	C
Prelude	CORES	N	Y	P
STRIDE	Next Generation	Y	Y	P
System Builder	ATECH System Builder	N	Y	PC
Wizard	Wizard Software	N	N	P

Table 2 3 *Applications generators available for PICK in 1986*

The table shows the name of the applications generator, the supplier and three additional pieces of information. These are, firstly whether the applications generator is limited to a particular licensee's implementation of PICK, secondly whether the applications generator was demonstrated or presented at a seminar on 4GLs organised by the UK PICK Users' Association in March 1987, and thirdly whether the applications generator generates programs in the form of files of parameter records which are interpreted at run-time (P) or in the form of DATA/BASIC program code which can be compiled and run (C).

In choosing an applications generator for the development, a number of factors were considered.

- The applications generator should run on the widest possible range of machines.
- There should be the flexibility of generating code or running in parameter-driven mode.
- The applications generator must be available to run on IBM PC compatible microcomputers using PC PICK.
- A run-time licence should be available at a lower cost than the development licence, or the package should develop stand-alone programs.
- The cost of the applications generator should not add considerably to the cost of the software
- It should be possible to incorporate program code written in PICK DATA/ BASIC into applications generated by the applications generator.
- The applications generator should have a track record of being used to develop major systems.
- The applications generator should offer the maximum functionality to the developer

Given the limited time available for the research, a choice was made relatively quickly. The ability to see some of the applications generators being considered demonstrated at International Spectrum exhibition in September 1986 helped to make the decision to use System Builder. The presentations at the UKPUA seminar in early 1987 confirmed that the choice appeared to be correct. A further comparison of dıfferent PICK applications generators is provided by Samish (1988).

System Builder

System Builder was chosen for the followıng reasons.

- It offered the best performance and functionality for the prıce.
- It was already in use extensively.
- Although it had been limited to a particular manufacturer's computers, it had recently been made more widely available.
- The prıcing for PC and small systems was competıtive.
- It appeared to be easy to use.

Development environment

The software development was started usıng PC PICK R83 Release 1.0 and System Builder Release 3.0, and completed using software upgraded to PC PICK R83 Release 2.2 and System Buılder Release 4 2. The software was run on a Sperry PC IT (IBM PC/AT compatıble) mıcrocomputer.

Thıs confıguratıon was chosen to provide a relatıvely cheap development environment, with the abılity to transfer software to larger mını-computers for larger operators. The PICK Systems Inc ımplementation of PC PICK sets the R83 standard for the PICK operating system. While thıs versıon does not have some of the features of more advanced or sophisticated versıons of PICK running on some lıcensees' machınes, ıt provıdes a standard subset, partıcularly ın the DATA/BASIC language whıch should be portable across all versions of the operatıng system. The combinatıon of System Buılder and DATA/BASIC provıded the development speed of a 4GL combıned wıth the flexibılity of third generatıon language programmıng.

Thıs chapter has outlıned the approach whıch the research has taken, firstly ın terms of the community action based orientatıon, secondly ın terms of the collaborative role whıch CT operators played ın the software development, thırdly in terms of the systems analysıs tools and technıques applied, and fourthly ın terms of the software tools and languages used to develop the software. Chapter 3 represents the application of these approaches to the understanding of CT operations and the devlopment of an underlying data model to support the range of CT services. Firstly four of the packages developed for the CT market which were in use at the start of the research are analysed. In particular, E-R charts, described above, are used to show the file structures of these packages. This analysis of existing software highlights shortcomings in the data structures on which they are based. The data model for the MULTI TRIP software, based on the analysis carried out using the methods and tools described in this chapter, is then developed

Chapter 3

Data Models for Community Transport

This chapter is written in two parts: the purpose of the first is to provide a background for the second. The first part analyses the data structures of four packages which were available at the time that the research started, and which are typical of the software available for CT services. The second part presents a data model for the computerised operation of community transport services which can be used as the basis for the structure of a software package for CT.

Existing Approaches

In order to clarify the shortcomings of existing approaches, the file structures of two of the three packages commercially available in the UK in 1986, of one North American package and of one package developed by a CT operator in-house will be summarised, before proceeding to develop a data structure for an integrated model.

Fletcher's D-A-R

A summary structure of Fletcher's D-A-R is shown in Figure 3.1, in terms of data files, processes which use the files, and documents which act as sources for processes or which are produced by processes Figure 3 2 shows the file structure as an entity-relationship (E-R) chart. These charts are derived from user documentation and from the copy of D-A-R purchased by the IT in CT Project for its earlier research and installed at Nottingham Dial-a-Ride

The obvious shortcoming in the design of D-A-R is that it takes no account of the possibility of other services using the same vehicles, as it was designed for a single service dial-a-ride style of operation. There are a number of other points about the package which are worth highlighting in a summary of this nature.

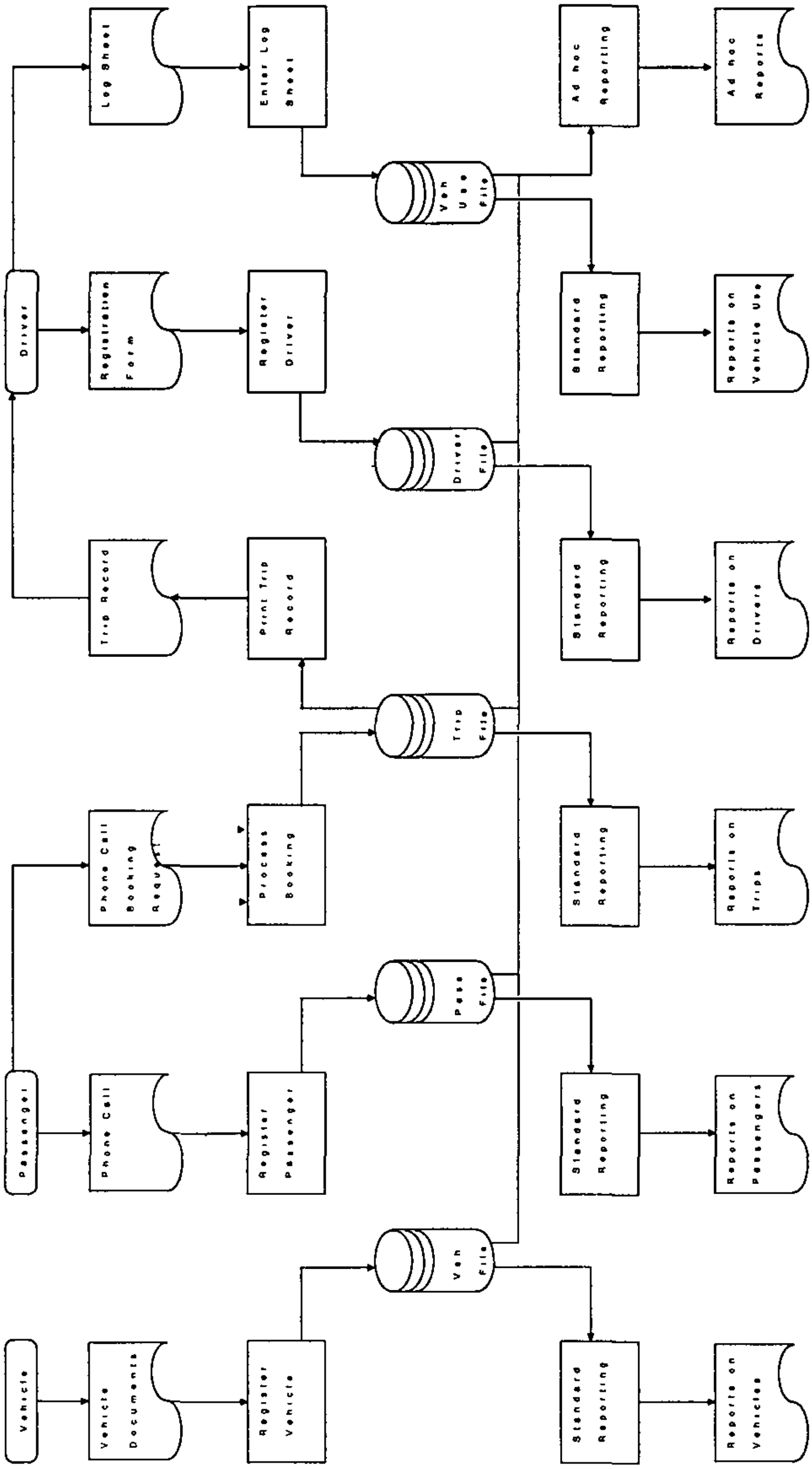


Figure 3.1 *Summary structure of Fletcher Computer Systems' D-A-R program*

- There is no facility to enter data about passenger trips ınto the system after they have taken place, for example to record actual times and mileages for comparison with booked or estımated ones.

- There ıs no shifts file, so shifts are entered by the operator on passenger tnp bookings, but cannot be validated in any way.

- Each passenger tnp record holds details of the driver who is to drive that passenger trip, although drivers are, in operational terms, assigned to vehıcle shıfts rather than to individual passenger tnps.

- The vehicle use file takes no account of vehicle usage for purposes other than dıal-a-nde.

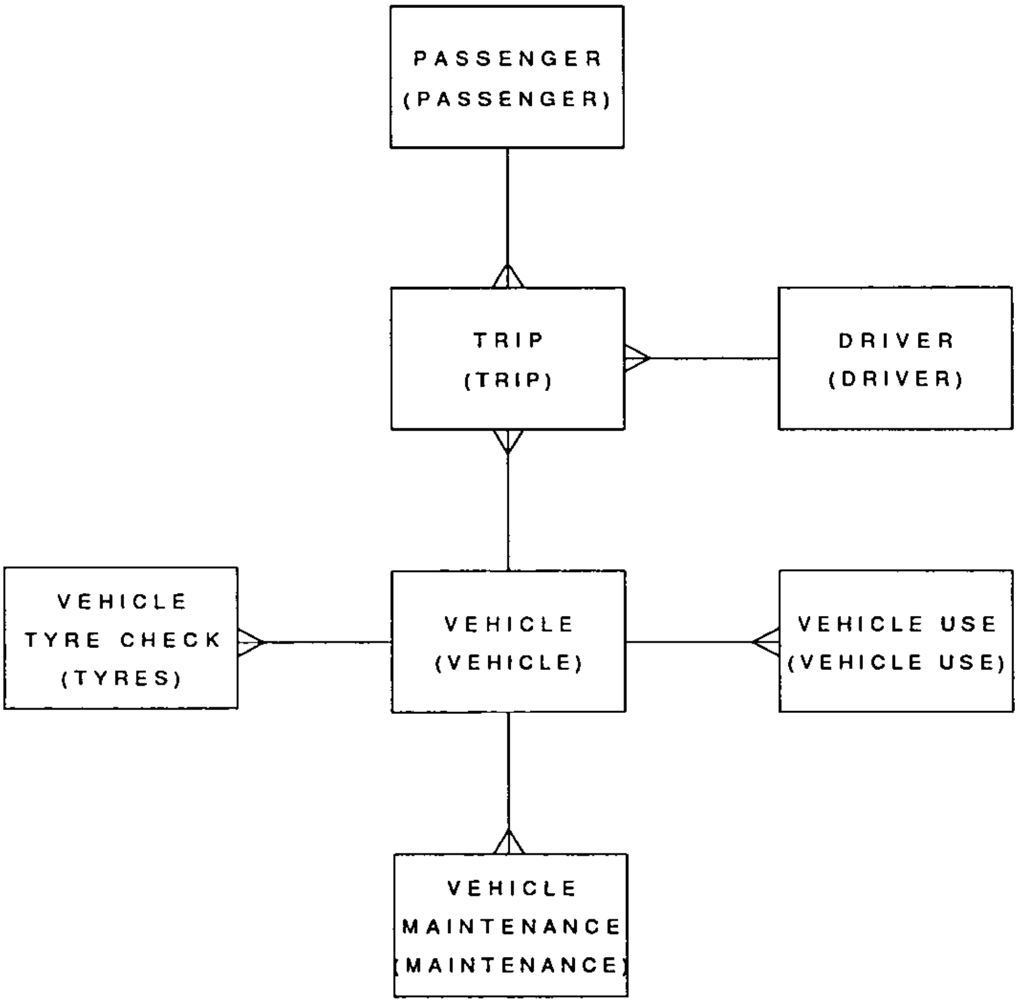


Figure 3.2 *Fıle structure of Fletcher's D-A-R as an E-R chart*

- Although not shown in Figure 3.1 or Figure 3.2, there is an accounts module included in D-A-R, however, this includes no links to the data in the rest of the package.

- To use the system properly, passenger records should be updated with details of the passenger's use of the vehicle; in Haringey, this is done by the despatcher using two terminals at once: one to enter the trip record and one to update the passenger record (Potter and Hagyard, 1986). This could be done automatically by the software.

- The fare structure is limited to a single fare rate for any distance, and a value must be entered into a fare table for each distance; concessions must be entered as fractional fares.

Perhaps the most serious shortcoming with the data structure of D-A-R is that there are no vehicle shift records and that passenger trips are allocated to vehicle shifts by operators with no facility to check on the actual availability of space on that vehicle at the times requested. The fleet number of the vehicle concerned is held on the passenger trip file, as is the shift code. To produce the trip records, in the form of schedules which are given to drivers and which form the basis for all vehicle shifts, the entire passenger trip master file must be searched to select trips relating to the date in question, and the resulting records must be sorted by vehicle fleet number, shift code and time. This has a number of implications.

- Firstly, it is time consuming, and is only normally carried out by D-A-R users at the end of a day, when all trips have been entered and any adjustments made

- Secondly, it requires that the passenger trip file should be cleared out regularly in order to allow the select and sort processes to run efficiently, and this means that data must be archived to a separate file for analysis at an early stage

- Thirdly, it means that there is no easy way of checking the status of the schedules in the computer without going through this time-consuming process, and thus no simple way of checking the proposed schedule in the computer system with the hand-written schedules used by the despatchers.

While some of the shortcomings highlighted above are not significant for users of D-A-R who only operate a single service, this last problem is one which affects all users, and is a result of inadequate data analysis at the design stage. It arises partly from the expectation that hand-written schedules would always be maintained by despatchers, that is that this process would never be computerised completely, and partly from a failure properly to analyse the relationships between vehicle trips and passenger trips, which can be provided by vehicle shifts. Although a Vehicle Shift Code is used in the Passenger Trip Booking Screen, this does not represent a real entity in the system.

ESCORT

A summary structure of TDD's ESCORT is shown in Figure 3.3 This is very rudimentary and is based on the available information TDD's brochure (Transport Design and Development, undated), and correspondence between Trisha Mannion of East Sussex County Council and Dr John Sutton of the Information Technology in Community Transport Project. The latter lists the available on-line functions as:

1. Book a Passenger Journey

2. Edit a Vehicle Journey

3. Delete a Vehicle Journey

4 Add a New Vehicle Journey

5. Delete a Passenger Journey

6 Make a Change to a Passenger Journey

7. Produce a Drivers Schedule or Transaction List

8. Display Report Menu

The former lists a subset of these functions, and additionally describes the function of 'Matching journeys to transport services' or AUTOSEARCH routine which attempts to find the most suitable vehicle service for a passenger journey.

Figure 3 4 shows the file structure of ESCORT as an E-R chart. This is derived from a document outlining the files and their contents (TDD, 1985)

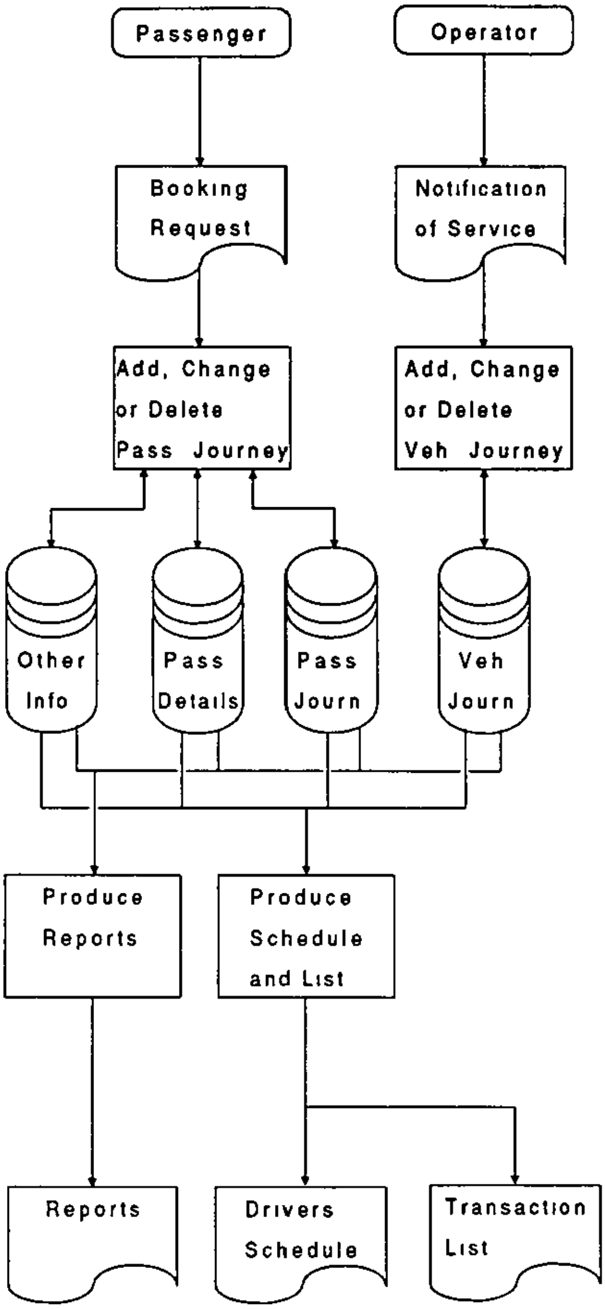


Figure 3.3 *Summary structure of ESCORT software*

ESCORT is very similar to Fletcher's D-A-R, in that it is designed for a single service type of operation, and makes no attempt to incorporate any features which might be of use in other services. This is not so much a criticism, as a reflection of the purpose and market for which it was designed. Certain points are worth highlighting.

- A relationship is built in the data between Vehicle Journeys and Passenger Journeys, however this is implemented by linking the first Passenger Journey on a Vehicle Journey to the Vehicle Journey and then linking all subsequent Passenger Journeys in a linked list structure.
- The file design seems to reflect the constraints of FORTRAN more than operational requirements.
- Although charges may well be allocated to County Council Departments rather than paid by passengers in fares, there is no facility other than a reporting option as part of the statistical analysis for dealing with this.

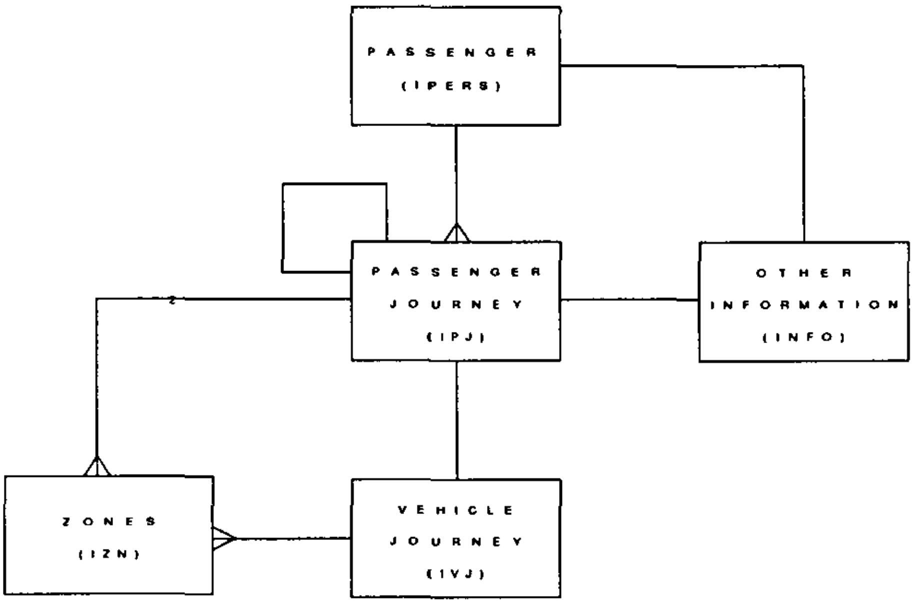


Figure 3.4 *File structure of TDD ESCORT as an E-R chart*

TRANSIT

A summary structure of TRANSIT is shown in Figure 3.5. Figure 3.6 shows the logical file structure as an E-R chart. These charts are derived from a demonstration copy of TRANSIT and the user documentation supplied with it (Giangrande, 1985). The physical file structure of TRANSIT is quite different from the logical file structure, in that a number of instances of entities that are represented by files in the logical structure are represented by sets of files in the physical structure, with a file for each record.

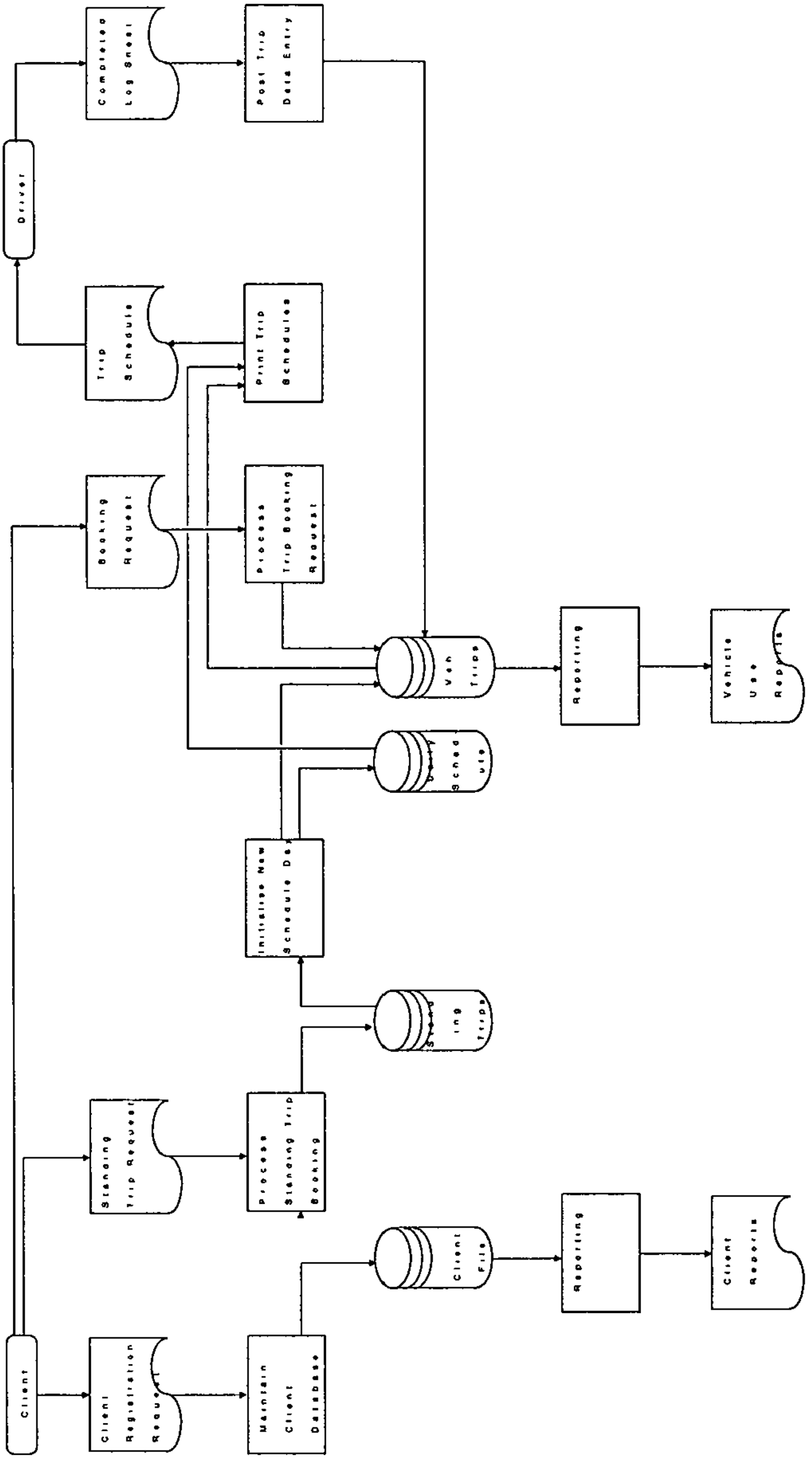


Figure 3.5 *Summary structui e of TRANSIT*

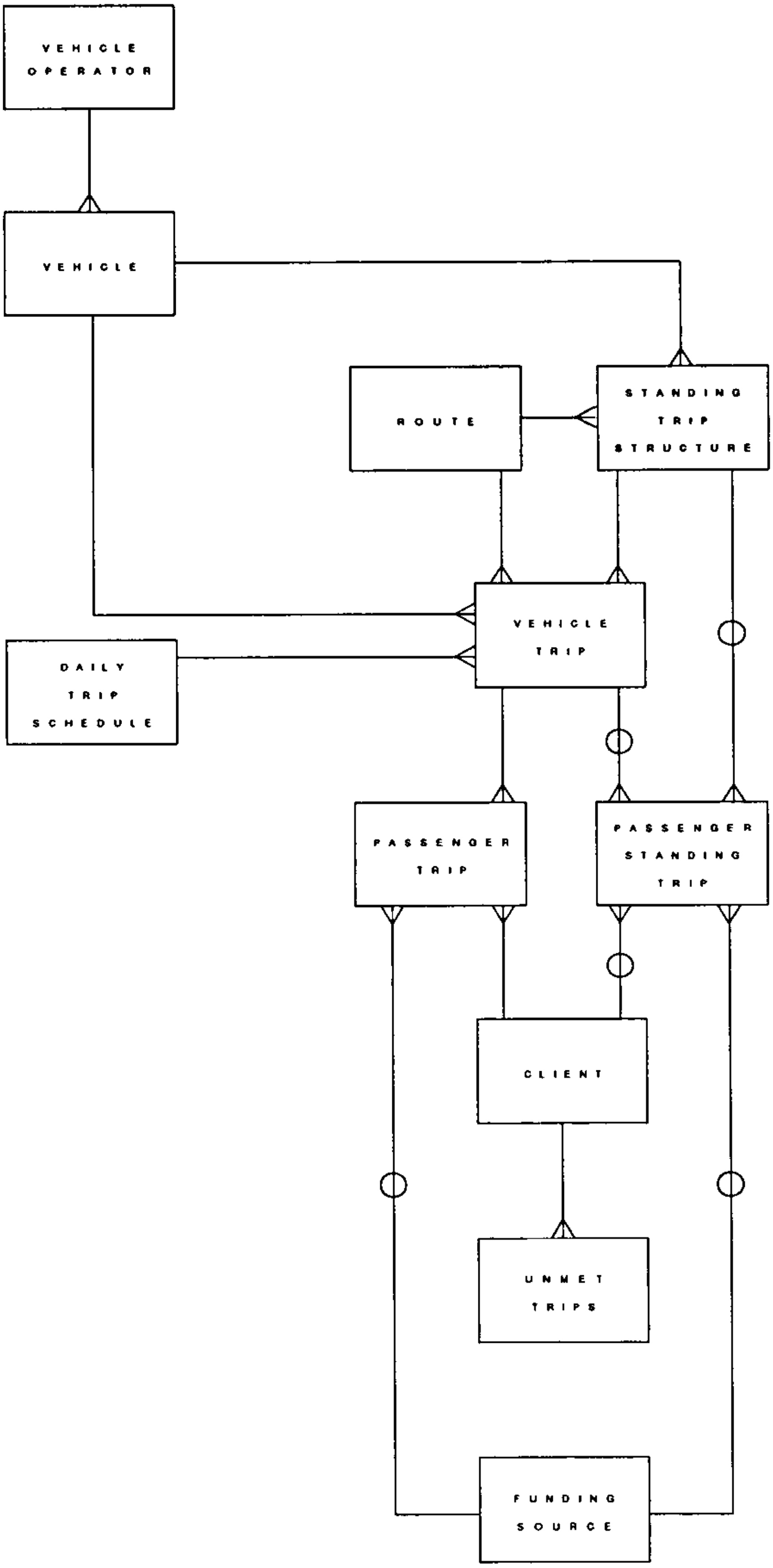


Figure 3 6 *File structure of TRANSIT as an E-R chart*

Like the previous two packages, TRANSIT makes no provision for other services, or for use of vehicles by other services. Other points worth noting are as follows.

- There are no facilities to maintain records of drivers who drive for the service.
- Although there are facilities for post-trip data entry relating to passenger trips, there are no fare calculation facilities.
- TRANSIT distinguishes clearly between vehicle trips and passenger trips, and has facilities to maintain records of regular passenger trips as part of records of regular vehicle trips.
- Passenger trips are entered into vehicle trip schedules, and it is possible to check spare capacity at a glance
- Vehicle trips are fixed in time and space. The system is designed to deal with vehicle trips which run from one zone to another at set times. In this sense it is closer to the dial-a-bus style of service in the UK.
- The system includes rudimentary routeing capabilities, using the space-filling curve algorithm (Bartholdi et al , 1983), although this was dropped in a subsequent version of the software, known as SST3 (Time Capsule, 1988).

Of the three packages discussed so far, TRANSIT provides the most comprehensive facilities, including regular vehicle and passenger trips, scheduling, routeing and post-trip data entry The subsequent rewrite of the software to run under dBase III has extended the usefulness of the original product which was designed as a demonstration program In the case both of TRANSIT and of SST3, the package is provided with full source code, which would enable anyone capable of programming in dBase II or III respectively to customise the application.

Derby CT Software

The Derby Community Transport software was written by one of Derby CT's volunteer drivers and modified over time in response to the needs of the organisation. The file structure is shown as an E-R chart in Figure 3.7. The functions provided are shown in Figure 3.8. The information on this package is derived from Ford and Forkin (1987) and the author's experience of using the

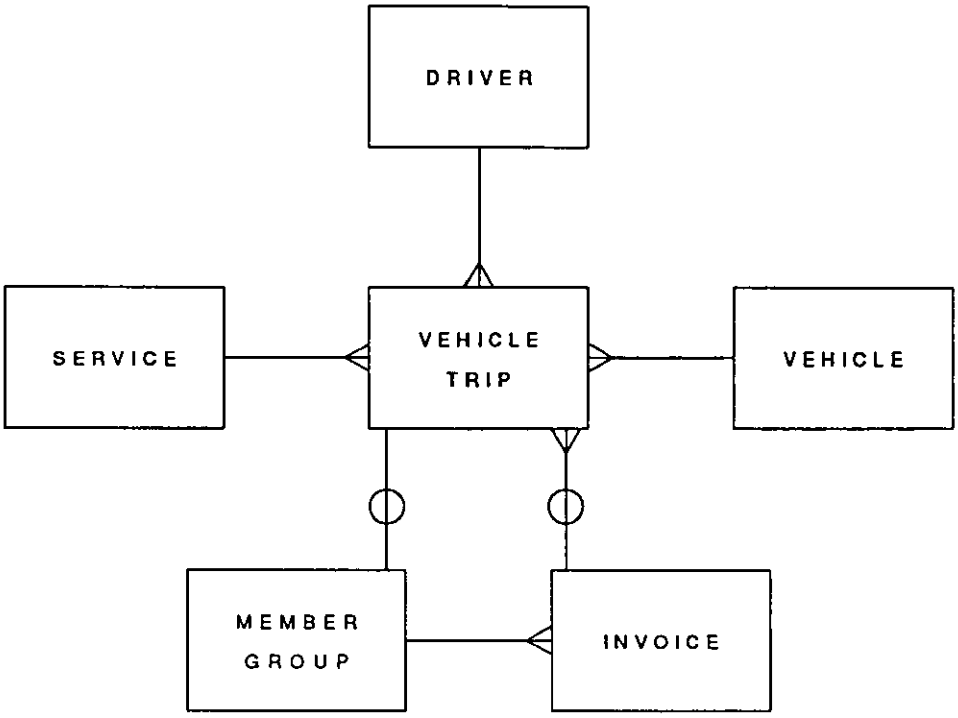


Fıgure 3.7 *Fıle structure of Derby CT software as an E-R chart*

package when employed by Derby CT.

The Derby software is probably the sımplest of the packages surveyed in terms of ıts file structure, and ıt does not cater for any advance entry of bookıngs. However, ıt ıs unıque among the packages surveyed ın that it caters for all three of the servıces provıded by the organisation usıng it - dıal-a-rıde, dıal-a-bus and group hire - as well as maintaınıng records on the organisation's own use of ıts vehicles, for example for traınıng or maıntenance purposes. Other poınts worth noting are as follows.

- The package ıs desıgned around vehicle use, and the post-trıp entry of logsheets is central to ıts function
- The package ıs closely tied to Derby CT's style of operatıon, as one might expect ın bespoke software.
- The maın purpose of the software is to produce ınvoices for users of the group hıre service, the production of statistıcs on that and other services ıs to some extent secondary.

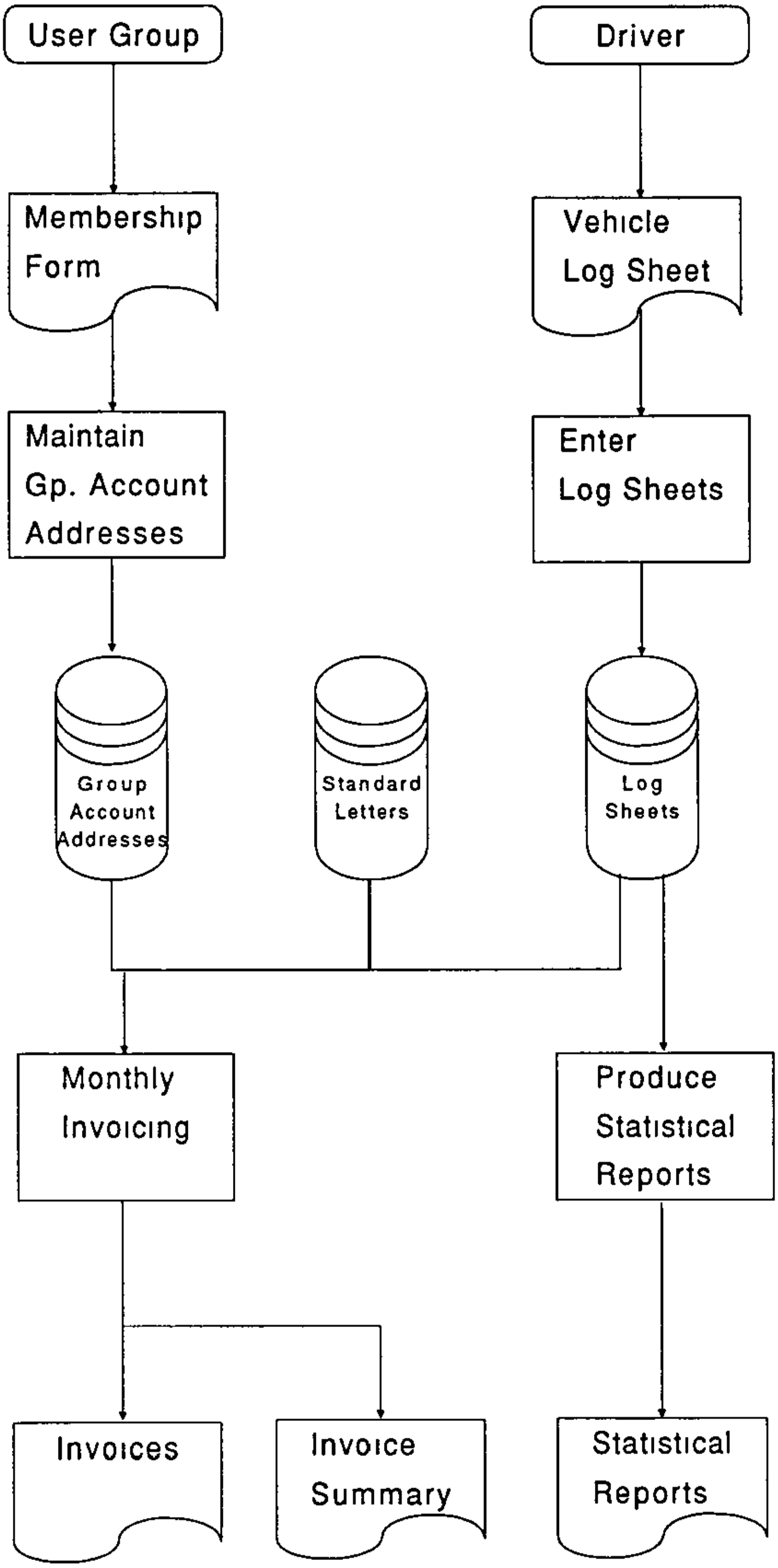


Figure 3.8 *Summary structure of Derby CT software*

- The calculation of invoice costs is heavily hard-coded into the software, and would not be easy to change for another operator with a different approach to charging.
- Although the financial requirements of invoicing are central to the design, there is no provision to maintain customer accounts or records of payments.
- A mailing list was also maintained using Delta on the PC, but no attempt was made to integrate the two packages, which held many addresses in common.

The software was originally written using the Delta database management tool, but was subsequently largely rewritten in BASIC to provide better validation of data entered into the system

Summary of existing packages

Each of the four packages analysed in this chapter has certain strengths and weaknesses Elements of each could find a place in a package designed for use by CT operators providing multiple services. The following requirements arise from this comparison and identification of shortcomings.

- It is necessary to distinguish between passenger trips and vehicle trips for services which provide transport for individual passengers.
- A single trips file can provide the basis for all services.
- A common logsheet can act as a data entry document for all services.
- For dial-a-ride, a vehicle shifts file allows checking of vehicle capacity to be carried out.
- For each service, provision can be made for the entry of bookings prior to trips, and the entry of post-trip data.
- Where names and addresses are held in a separate file, these should be integrated into the rest of the system
- Where accounting information is held, this should be integrated into the rest of the system.

In the rest of this chapter, these points are taken into consideration in developing the data model for CT operations. However, the development of the data model is also rooted in an analysis of the existing manual procedures and records in the Collaborative Projects.

The Data Model

The detailed analysis and documentation of the data model for CT fills several ring-binders. In order to explain how the model was derived and to keep the scope of examples within a reasonable range, the following description will concentrate on the design of three elements of the system, before explaining the overall model.

This is presented in stages which reflect the analytical process which was undertaken during the research, and which use the techniques described in Chapter 2. However, these stages are inevitably summarised and idealised to reflect the constraints of presenting this information in a readable and understandable form The stages presented here are as follows.

1. Narrative description of group hire operation
2. Narrative description of vehicle brokerage variation of group hire
3 Analysis of entities in group hire operation.
4. Analysis and integration of entities involved in vehicle brokerage
5. Narrative description of dial-a-ride operation.
6. Analysis and integration of entities involved in dial-a-ride operation.
7. Identification of common entities

Each is explained in turn below.

Narrative description of group hire operation

A community transport operator (operator) makes vehicles available to member groups for hire The operator is typically a charitable organisation, constituted as a voluntary group. The vehicles may be owned by the operator, or they may be leased, possibly from a local authority. Some or all of the vehicles may be adapted to carry passengers with transport handicaps, for example, by having ramps or tail-lifts fitted and through the ability to carry passengers in wheelchairs. The vehicles are normally of a size which is covered by the Minibus Act 1977, which can be driven without the need for a PSV licence.

Member groups are typically voluntary, community or charitable organisations. Membership may be by an annual fee, or it may be available on a trip by trip basis. Vehicles are not normally available for hire to organisations which are not in some way voluntary or charitable or to individuals.

Member groups may nominate individuals to drive the vehicles for them These drivers are normally subject to some kind of program of training and testing, both in vehicle handling and safety and in dealing with passengers with disabilities. Individuals may also volunteer to drive for the operator as pool volunteers without necessarily having any allegiance to a particular member group In some cases, operators are able to employ drivers on a full-time or part-time basis.

Member groups book the use of vehicles in advance. They may be able to make regular or block bookings, for example, for regular luncheon clubs or other events. Many operators require that any person making a booking on behalf of a member group should be authorised by that member group as a valid person to make bookings.

When a booking request is received, a number of checks may be made: the status of the group making the booking is checked (Is the membership up to date? Does the group owe any/much money for previous bookings?); the status of the individual making the booking is checked (Is he/she authorised?), the availability of a suitable vehicle or vehicles for the date or dates and times requested is checked; the availability of a driver or drivers is checked.

A booking may be for a simple trip or a more complicated set of trips. It may consist of a single round trip, during which the group retains use of the vehicle. The trip may be contained within a single day, or it may run over more than one day. The booking may involve more than one vehicle, if the number of passengers to be carried exceeds the capacity of a single vehicle or of a single available vehicle. It may be necessary to reallocate existing trips to accommodate the new booking more efficiently. The booking may involve an outward trip or trips at one time after which the vehicle or vehicles are free for other purposes, and a return trip or trips later the same day or another day. The trip or trips may involve single pick-up points and destinations or may involve multiple pick-ups of passengers from a list In some cases, this can be left to the member group to organise, in other cases, for example if a pool or employed driver is to drive, then the operator requires details of the passengers' names and addresses.

In some cases the trip or trips may not be allocated to specific vehicles at this stage, but the decision may be taken later when a better picture of demand is available

A driver or drivers may be allocated to the trip or trips at this stage, particularly if nominated by the member group, or if the booking is for trips during the day time, and the operator employs drivers on set shifts. It may be necessary to take the booking and to find a suitable available volunteer driver at a later stage.

A booking record may be produced and sent to the member group for signing and returning, possibly with a deposit

If the operator cannot meet the request for the booking, a record may be maintained of the refusal and the reasons for it, for the purposes of shortfall monitoring

On the date of the trip or trips, details of pick-ups, passenger lists and destinations must be given to the driver with the vehicle keys This may be in the form of a single job sheet with all the required details on it. For evening trips, when the office is not staffed, arrangements may have to be made to get these to the driver during the day.

The driver of each vehicle may be required to complete a vehicle log sheet with a line of detail for each trip, or the details may be entered on the job sheet for the booking, or it may be necessary to complete both. Job sheets would be returned to the office with the keys, log sheets would be left in the vehicle until completed. The driver may additionally provide the operator with information about accidents or damage to the vehicle during the trip, records of fuel put in the vehicle during the trip, and records of expenses claimed by him or her.

The information from the log sheets or job sheets forms the basis for billing the member group. The bill may be calculated in a number of ways, based on time or mileage or both. Fuel, damage and drivers' expenses may be added to the bill.

Narrative description of vehicle brokerage variation on group hire

In a vehicle brokerage operation, some or all of the vehicles made available for hire to member groups by the operator are owned by organisations other than the operator, and used by those groups for their own purposes. These groups may or may not also be member groups and thus users of vehicles in the scheme. When the vehicles are not being used by their owners, they are available for use by other organisations, typically the member groups of the community transport operator.

There are three possible levels of integration of these vehicles into the operator's administrative procedures, and a fourth which integrates the vehicles into the operator's fleet.

The first level is that of information only. The operator, as broker, holds information about the vehicles and their general availability. If the operator is unable to meet a member group's request for a booking, the member group is provided with the information about the vehicles in the brokerage scheme and it is then up to the member group to make a booking with the owning organisation

The second level is that of bookings only. The operator, as broker, is able to take bookings for vehicles owned by other organisations in the brokerage scheme, and is therefore provided with full details of the owning organisation's intended use of its vehicles. However, the financial arrangements are between the member group and the owning organisation and do not concern the operator

The third level reflects total integration of the brokerage vehicles into the operator's administrative procedures. The operator handles all bookings and bills member groups for use of brokerage vehicles. The income from hire is then passed on to the owning organisations, usually subject to a charge by the operator to reflect the administrative costs of performing the brokerage.

The fourth level integrates the brokerage vehicles into the operator's fleet and fleet management. The operator may be responsible for the garaging, maintenance, tax, insurance and other overhead costs of the vehicles in the brokerage scheme, and these will be charged on to the owning organisatıon.

Vehicle Brokerage impinges on the booking process in two maın ways. Fırstly, the operator may accept bookings wıthout being able to allocate a vehicle to the booking, and will check the availability of vehicles with owning organisatıons before confirming or refusing a bookıng. Secondly, constraints may be imposed on the type of passenger who may be carried on vehicles owned by certain organisatıons. For example, the constitution of a charity for elderly people may mean that vehicles which it owns can only be used in line with its charıtable purposes, and thus only to carry elderly people. Such vehicles would not be avaılable for hire through brokerage to youth groups.

The ıntroduction of the third or fourth levels of brokerage also requires the operator to maıntaın fınancıal records of dealings wıth vehıcle owners as well as with member groups, for example remittances for hire income received or charges made for fleet management functıons performed

Analysis of entities in group hire operation

Readıng through the narratıve descrıption of a group hıre operation, a number of key entıtıes, wıth whıch the operation is concerned become apparent. These entıtıes are lısted ın Table 3 1

The inıtıal work of the research was based around the lıst of proposed files shown ın Table 3.2, whıch was compıled by the original User Group. A feature of both lısts is that they ınclude as entıtıes or fıles objects which subsequent analysıs reveals to be documents. For example, a job sheet ıs a prınted record of all or part of the detail of a booking, whıle a logsheet ıs an input document containıng data about indıvidual trips which will be entered ınto the system. However, the list in Table 3.2 also reflects an attempt to identıfy records to be held in computer files at too early a stage ın the analysis, and reflects a confusıon between documents and files and a failure to represent relatıons between entities correctly, for example by ımposing arbıtrary lımıts on the number of drivers for a particular member group.

Entities
operator
vehicle
member group
passenger
driver
booking
regular booking
authorised contact
trip
booking record
refusal
job sheet
logsheet
damage record
expenses record
fuel record
bill

Table 3.1 *Entities involved in group hire operations, from the narrative description*

File Description
Group Membership Record
Mailing List
Driver Record
Driver Test Record
Vehicle Permanent Record
Vehicle Daily Record
Vehicle Abuse Record
Accident Record
Maintenance Record
Booking Record (Pre-trip)
Vehicle Journey Record
Passenger Journey Record
Group Passenger List
Group Regular Trip Pattern

Table 3.2 *Files in the User Group's original specification*

The use of entity-relationship analysis helps to clarify the status of the entities listed and the relationships between them. Figure 3.9 illustrates the relationship between the entities 'operator', 'member group', 'vehicle' and 'driver' taken from the list in Table 3.1, and introduces the additional relationship between drivers and member groups. Figure 3.10 shows the successive refinements of the relationship between member groups and their use of vehicles, from the simplistic 'member group books vehicle' through to the more complex representation of the system involving authorised contacts and vehicle trips. Figure 3.11 integrates the two sets of relationships from 3 9 and 3.10 into a single E-R chart.

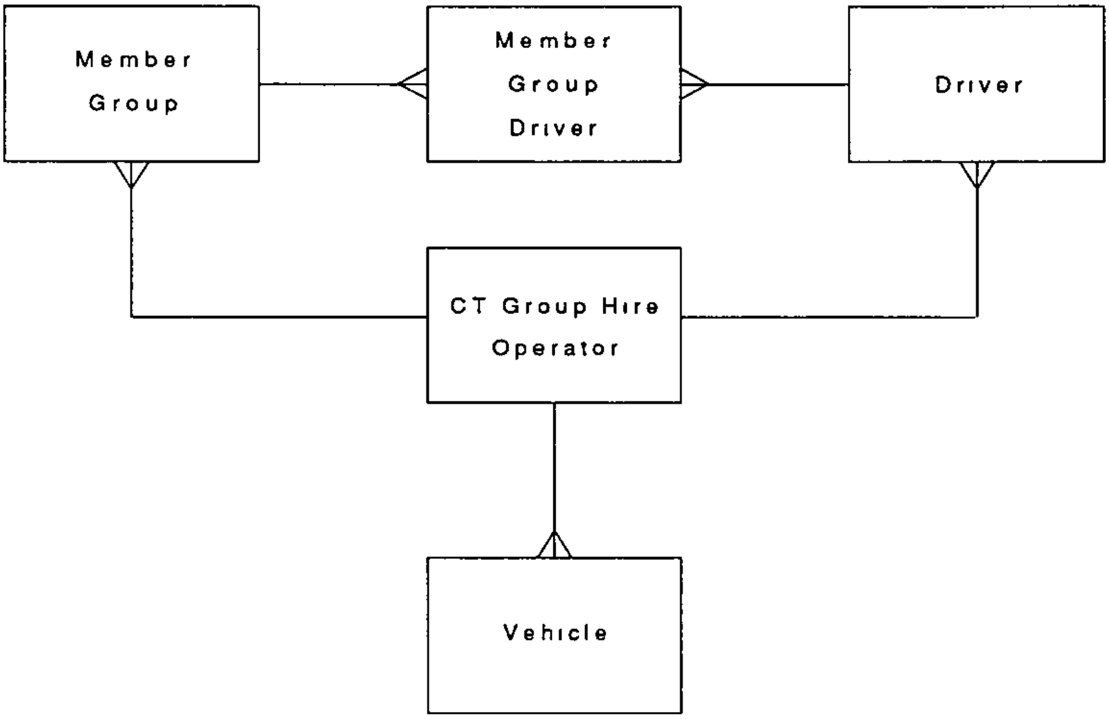


Figure 3.9 *Relations between operator, member groups, drivers and vehicles*

The introduction of the idea of regular bookings makes it necessary to distinguish in some way between a booking which is going to occur on a particular day or days and a regular booking which has the same relationships but acts as a pattern from which actual bookings are generated. The use of the bookings diary provides this link to a fixed date or dates, and this is shown in Figure 3 12. The diary is explained in detail in Chapter 4

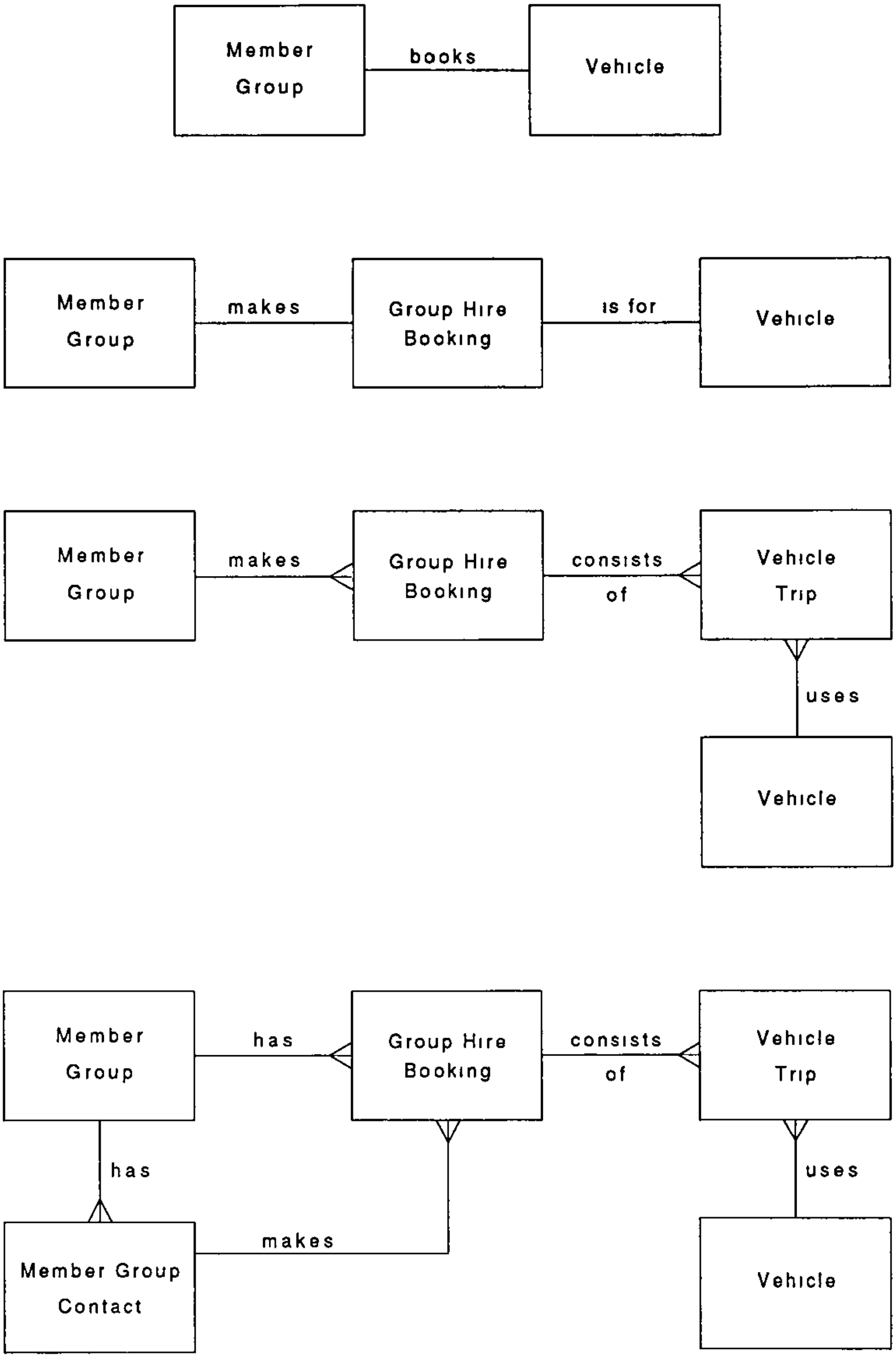


Figure 3.10 *Successive refinements of group hire bookings*

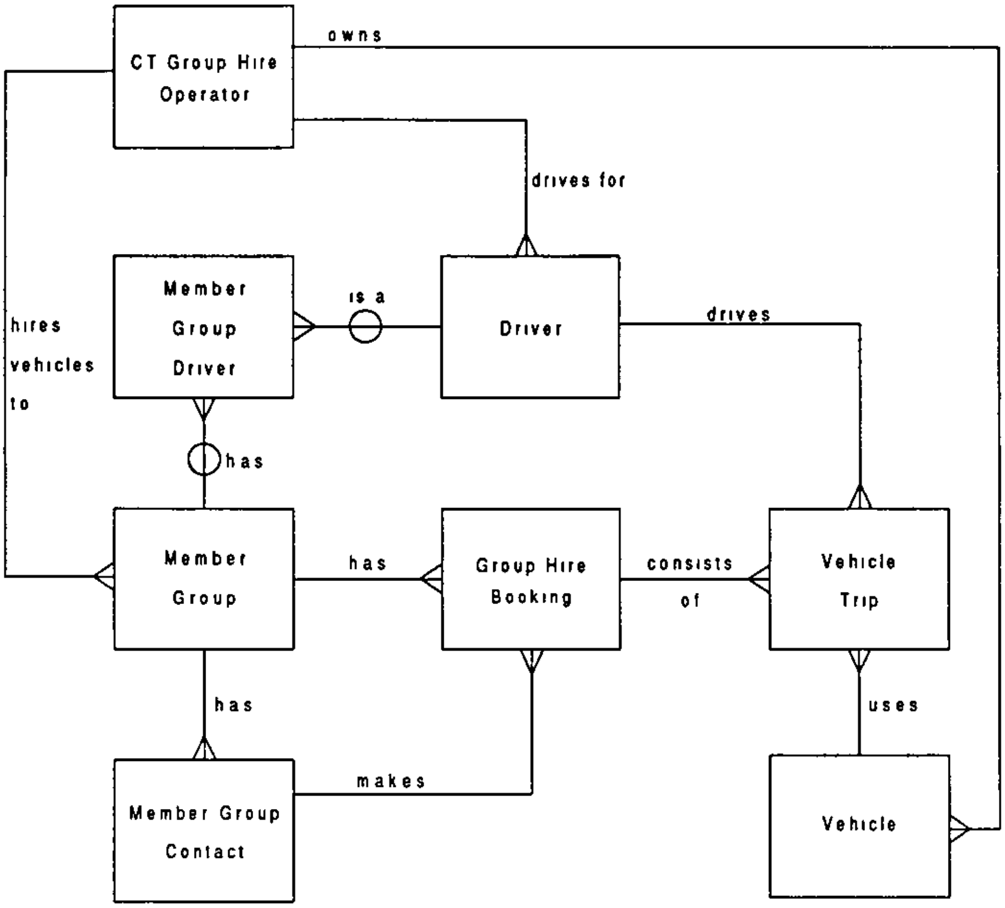


Figure 3.11 *Integration of group hire bookings with group hire operator, drivers and vehicles*

Finally, it is necessary to account for the cost of vehicle trips and the associated expenses, such as fuel, damage and drivers' expenses. Figure 3.13 shows the entities involved in charging for a booking. However, although some operators do send individual invoices to member groups, member groups do not always pay for individual invoices, but may pay for more than one at a time, effectively paying off an account Operators may also make their own transactions against such an account, raising credit notes or making adjustments, and it is necessary to introduce the additional member group account and group account transaction entities. Figure 3.14 is an attempt to combine all these entities and to add passenger lists and passengers into the scheme.

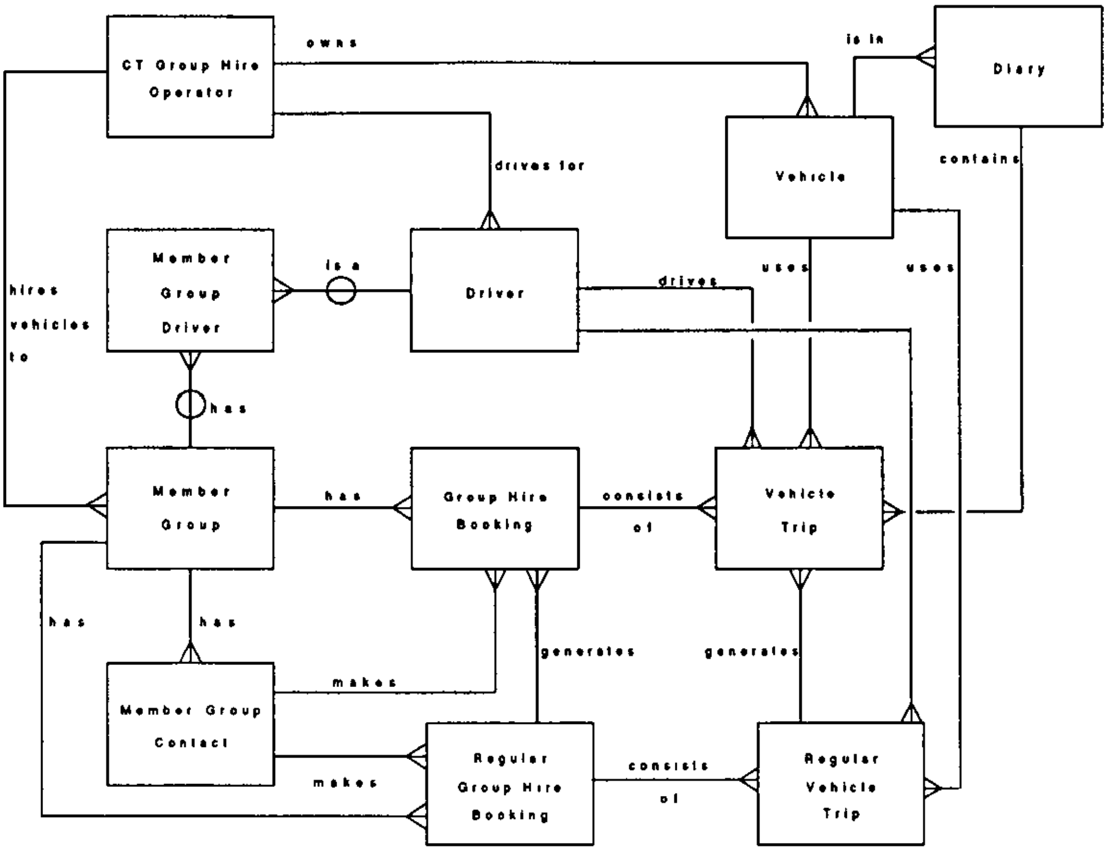


Figure 3.12 *Integratıon of regular bookings and the ıntroductıon of the vehıcle diary*

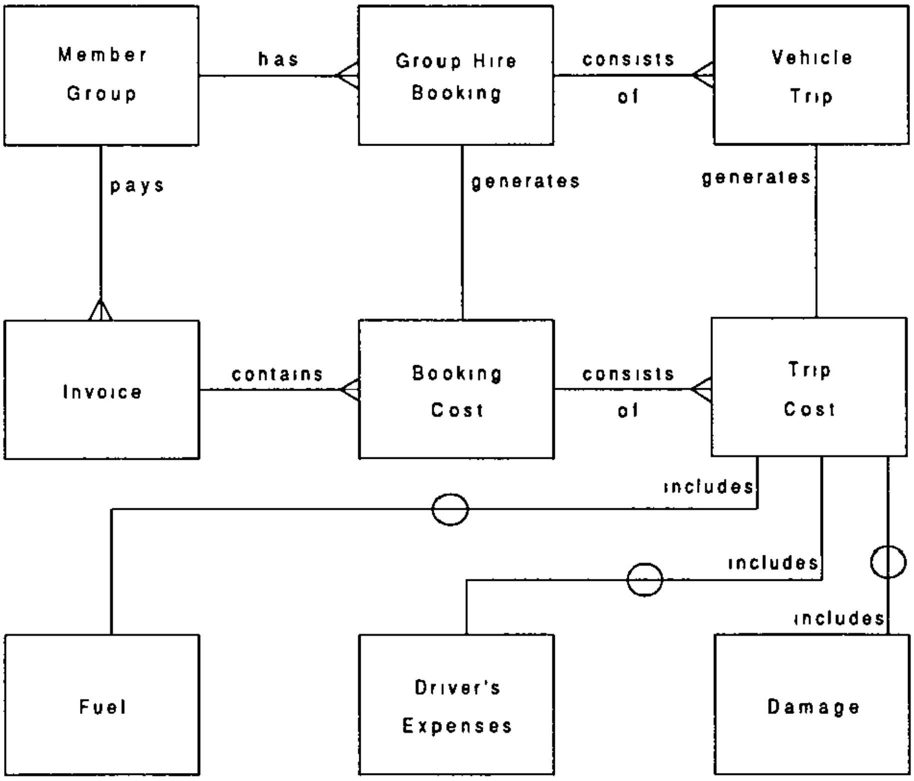


Figure 3.13 *Accountıng for group hire bookings*

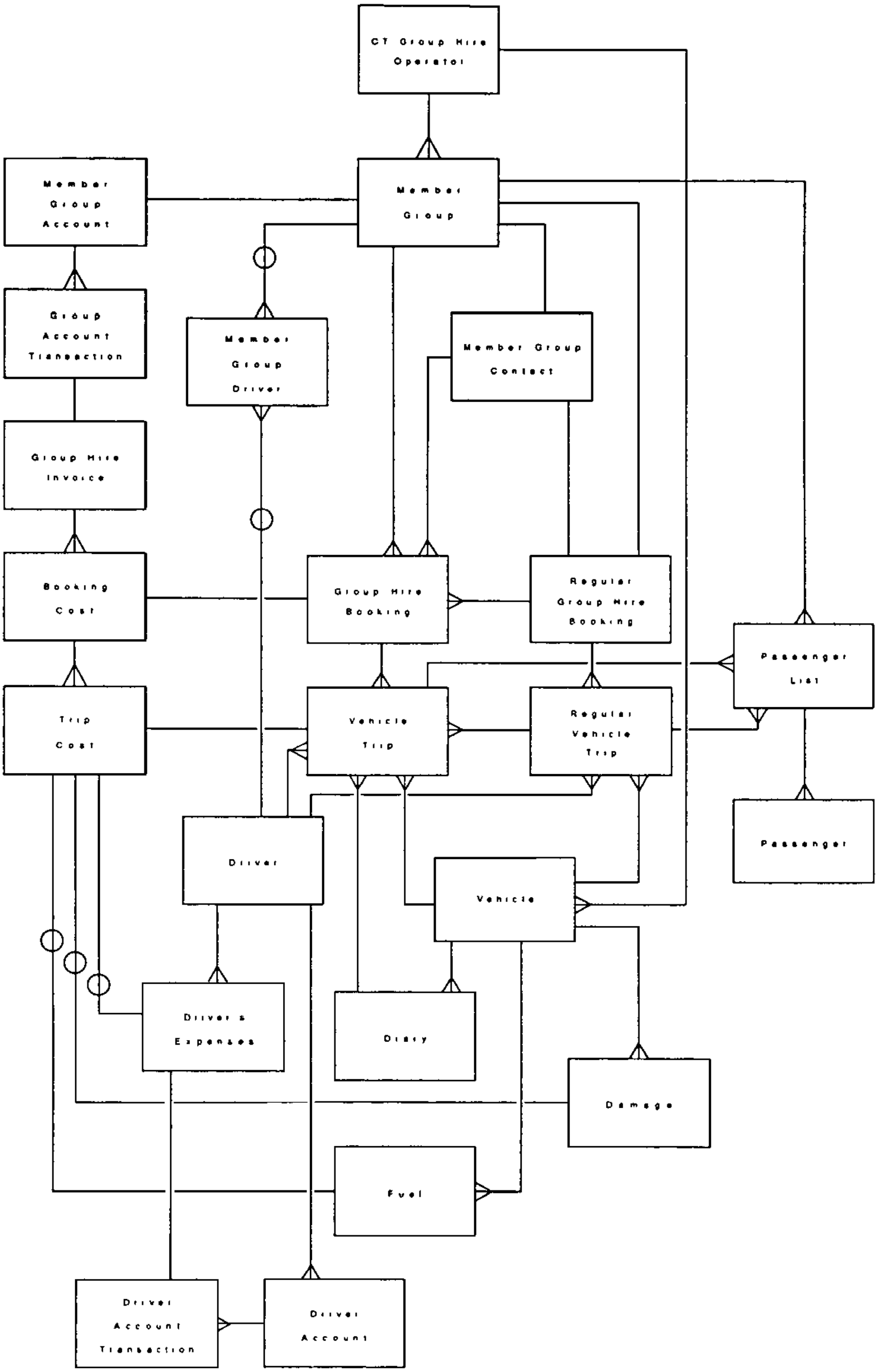


Figure 3.14 *Complete chart for group hire in isolation from other services*

Before moving on to examine the impact of the introduction of vehicle brokerage into this model, it is worth examining some of the processes which involve the entities in this model. Figure 3.15 shows a summary data-flow diagram of a group hire operation. Data-flow diagrams are hierarchical in nature, and can be successively decomposed into further levels of detail. The process in Figure 3.15 which is most significant is the box labeled '4. Enter and maintain bookings'. This process is concerned with the way in which group hire bookings are made and in particular with the way in which the validity of the bookings is checked. Figure 3.16 shows a document flow chart for the process of taking a booking based on the manual practices of Nottingham Community Transport, and recorded as a result of observation and interview in January 1987. Figures 3.17 and 3.18 show successively lower levels of the processing involved in entering and maintaining bookings.

Analysis and integration of entities involved in vehicle brokerage

The addition of vehicle brokerage into a group hire operation introduces additional entities into the model. The list in Table 3 3 was drawn up using the narrative description in the section above

Entities
vehicle operator
vehicle owner
hire remittance
management charge

Table 3.3 *Entities involved in vehicle brokerage*

Depending on the nature of the brokerage service offered by the CT operator, the owner of a vehicle can now be the operator of the CT service, a member group or another vehicle operator. These other operators could be other charitable vehicle operators or possibly commercial vehicle hire companies Figure 3.19 shows how these relationships can be represented The vehicle operator about whose vehicles information is held 'for information only' is not going to be involved in any booking

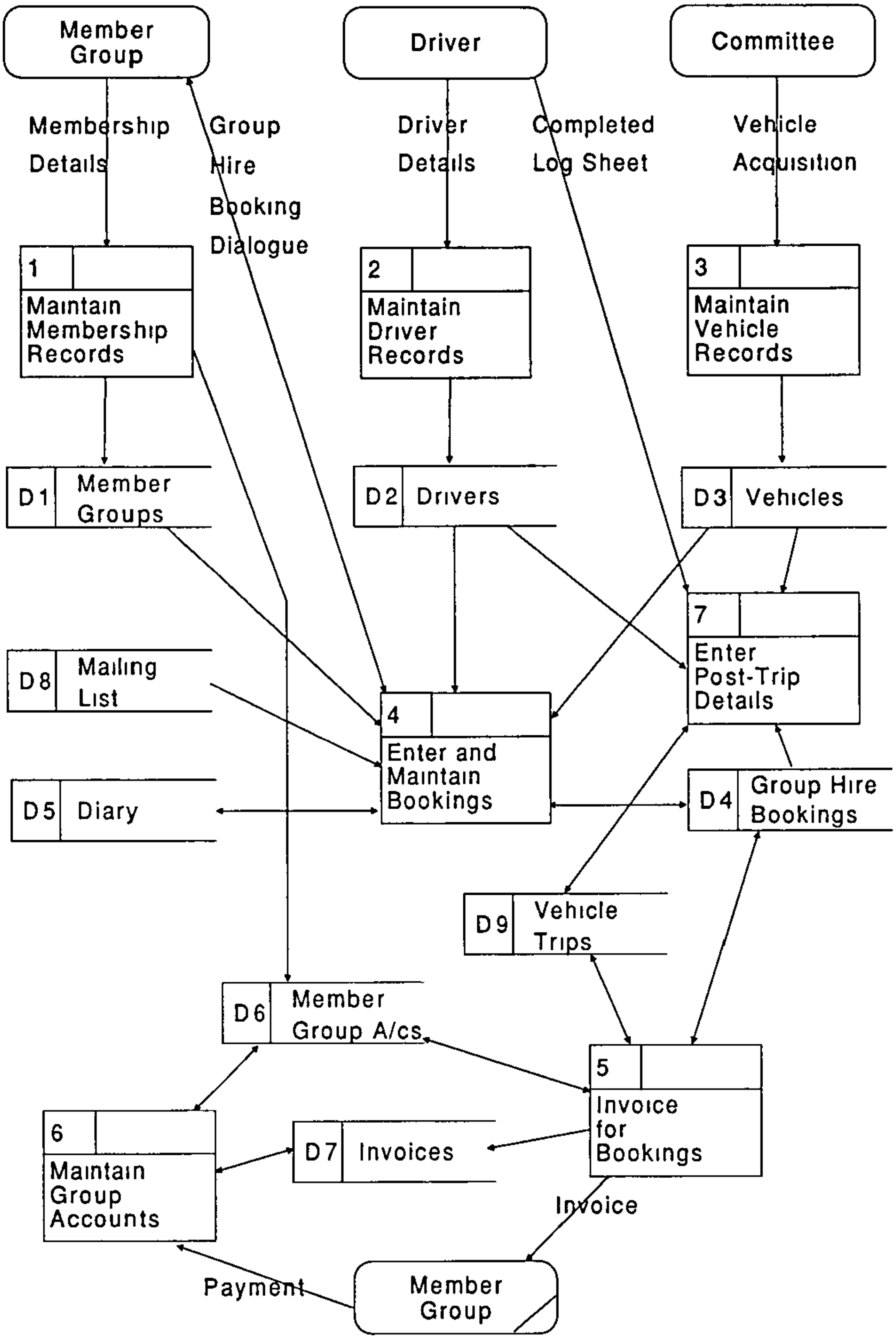


Figure 3.15 *Summary data flow diagram for group hire*

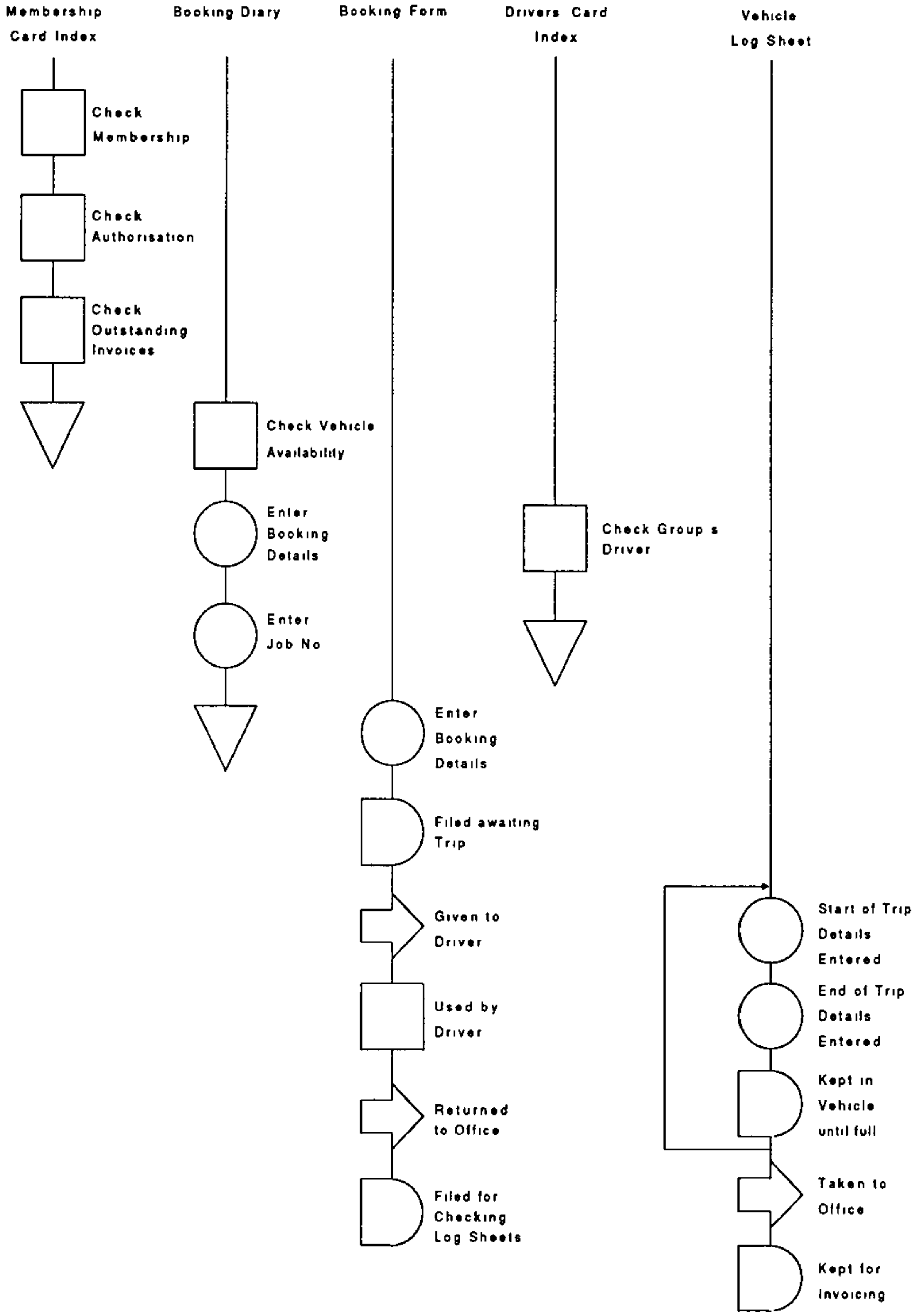


Figure 3 16 *Forms flow chart - Nottingham Community Transport group hire booking*

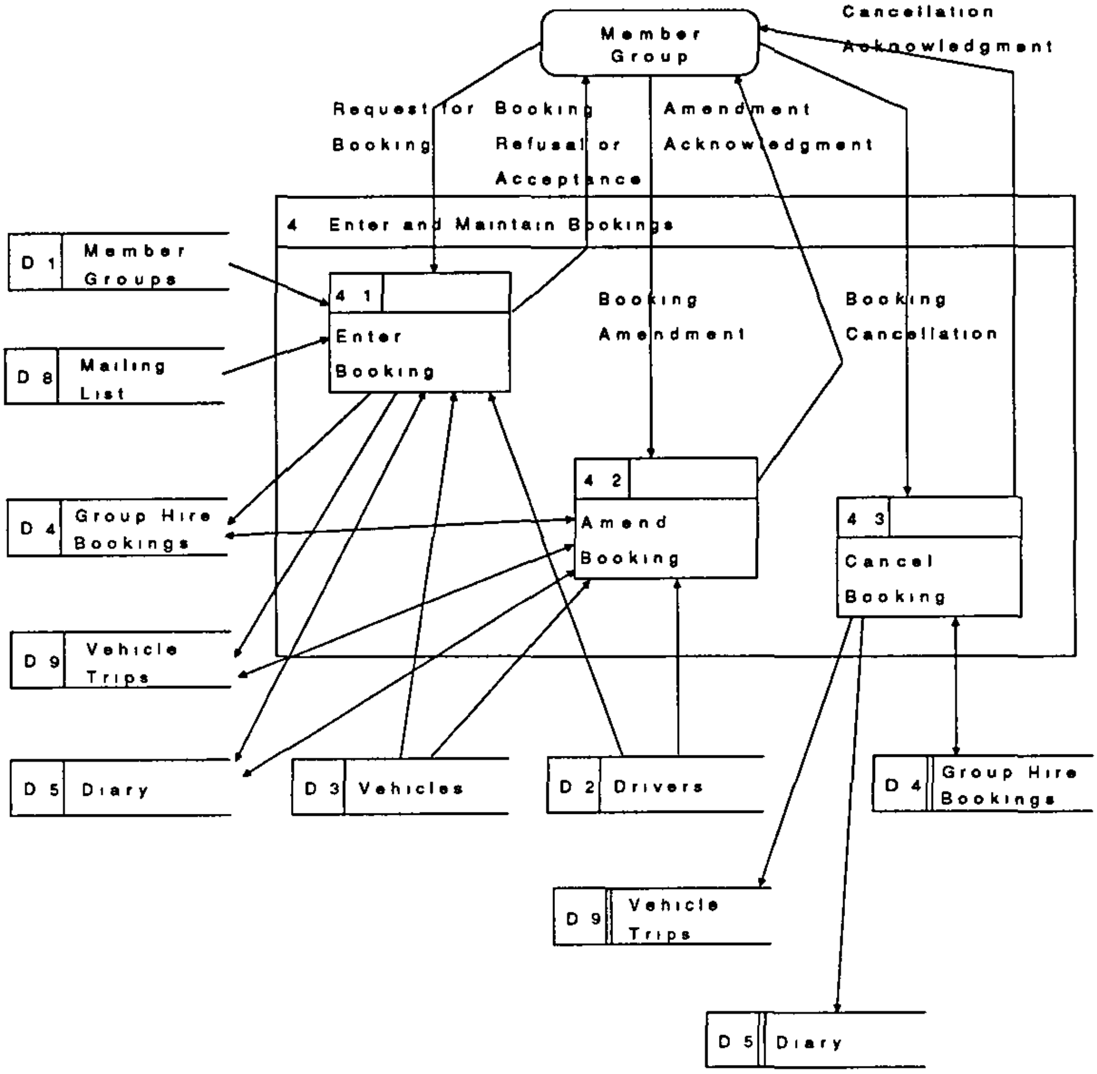


Figure 3.17 *Data flow diagram for Enter and Maintain Bookings*

processes or any accounting for vehicle usage, while any vehicle operator whose vehicles are booked out through the broker is likely to be involved in booking vehicles out for its own use and is therefore likely to be a full member of the scheme, and as such a member group. Thus in brokerage, the member groups file is used to hold groups which may be vehicle owners, vehicle users or both. This raises the question of the kind of brokerage operation in which the broker is not actually a vehicle owner, but only acts as a broker, and possibly uses vehicles belonging to member groups for its own purposes In this case, the CT operator is itself a member group, and not a separate entity in the model. In view of these

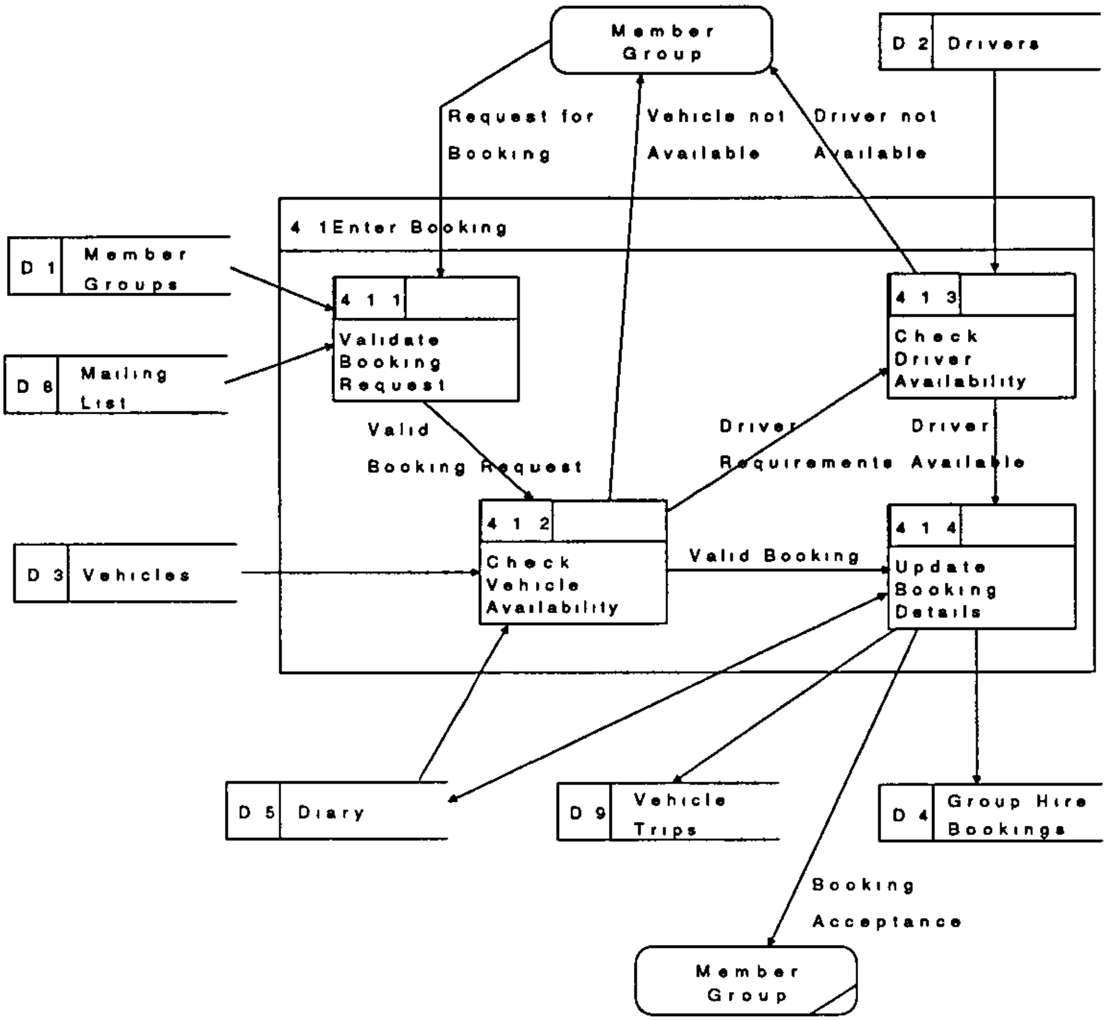


Figure 3.18 *Data flow dıagram for Enter Bookıng*

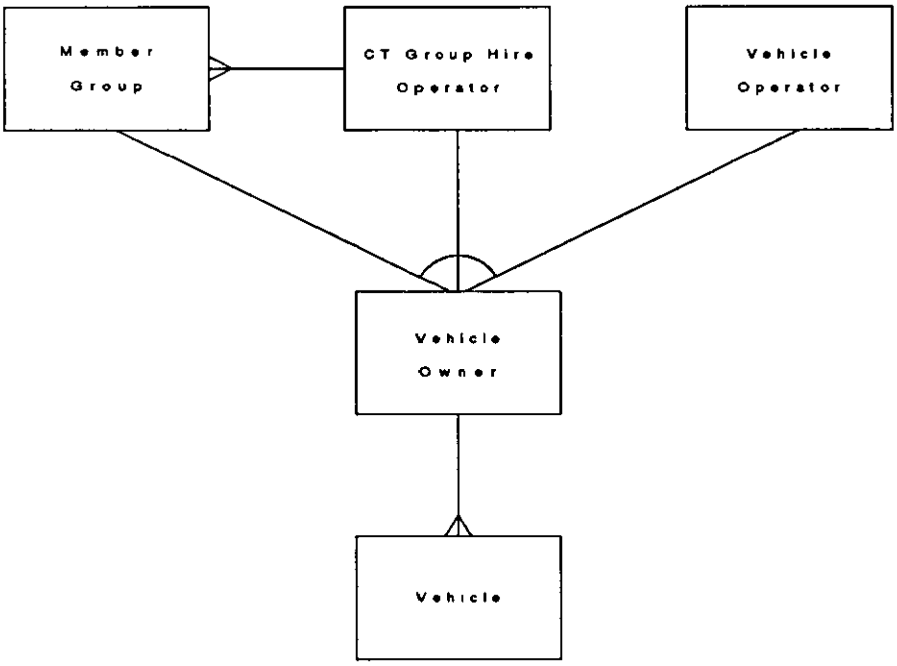


Figure 3.19 *Relationships ınvolving vehıcle ownership in group hire and brokerage*

considerations, it was decided to adopt a model in which the CT operator is itself a special case of the member group entity, and this is shown in Figure 3.20.

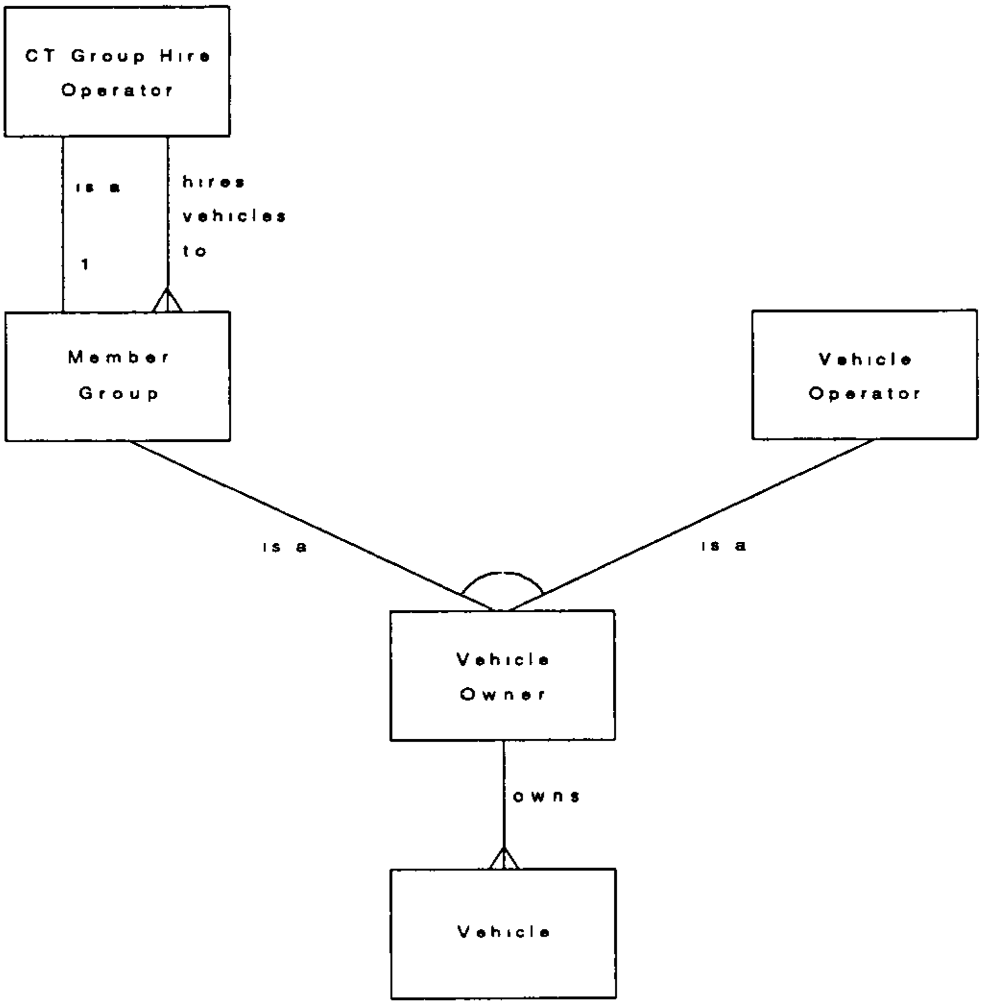


Figure 3.20 *Revised model of vehicle ownership in group hire and vehicle brokerage*

The involvement of other vehicle owners in the operation also highlights the requirement for an ability to record use of vehicles by those vehicles' owners for their own purposes. In the case of the CT operator, this could be use of vehicles in order to transport staff or volunteers, or in order to train or test drivers, or in order to maintain, service or fuel the vehicles. In the case of other vehicle owners, this could be any use of their vehicles which might be recorded by the broker either for statistical or analytical purposes or in order to maintain records of when the vehicles are unavailable for brokerage use. While it might be possible to include such vehicle use within the scope of group hire bookings, it was decided to separate it out as a different category of booking, own use bookings. The data model for own use bookings is similar to that for group hire shown in Figure 3.14. However, there is no requirement to account financially for the vehicle use. Figure 3.21 shows an E-R chart for own use bookings.

While there is no requirement to account for own use bookings, the introduction of brokerage into a group hire operation imposes on the operator the need to account for the use of vehicles which belong to member groups in order to pass on to those groups the appropriate hire fee for the use of their vehicles. If the vehicles are fully integrated, the operator may also be passing on to the owning groups the costs associated with managing the fleet and maintaining and servicing the vehicles. Thus any trip which is made in a brokerage vehicle will have an associated brokerage trip cost, which forms the basis of the hire remittance listed in the entity list for brokerage The brokerage trip cost may be calculated on a different basis from the operator's usual method of calculating trip costs, or may be based on the charge made to the group using the vehicle, less a flat rate or a percentage charge for brokerage. In some cases, the full charge for use of the vehicle may be passed on to the owning group, and a periodical management charge may be made by the operator for brokerage. These various requirements introduce entities associated with brokerage accounts into the data model. An account is created for any group owning a vehicle in brokerage, and all transactions relating to payments and charges for use of that owning group's vehicles are held in that account. Figure 3.22 shows the important entities involved in accounting for vehicle brokerage.

An addition to Figure 3.22 is the charge rate entity This is introduced in order to hold details of how different vehicles are charged for Some operators charge for different vehicles at different rates, some operators with brokerage schemes calculate the brokerage trip cost based on a different rate from that used to charge the user, and some operators have different rates for trips depending on criteria such as distance and length of time. As such, the simplistic relationship shown in Figure 3 22 does not represent the complexity of the situation, and Figure 3.23 more accurately represents the model

The additional introduction of the entities associated with brokerage into the data model highlights the role of the mailing list. The original concept of the mailing list was that it was no more than a list of addresses of individuals and organisations to

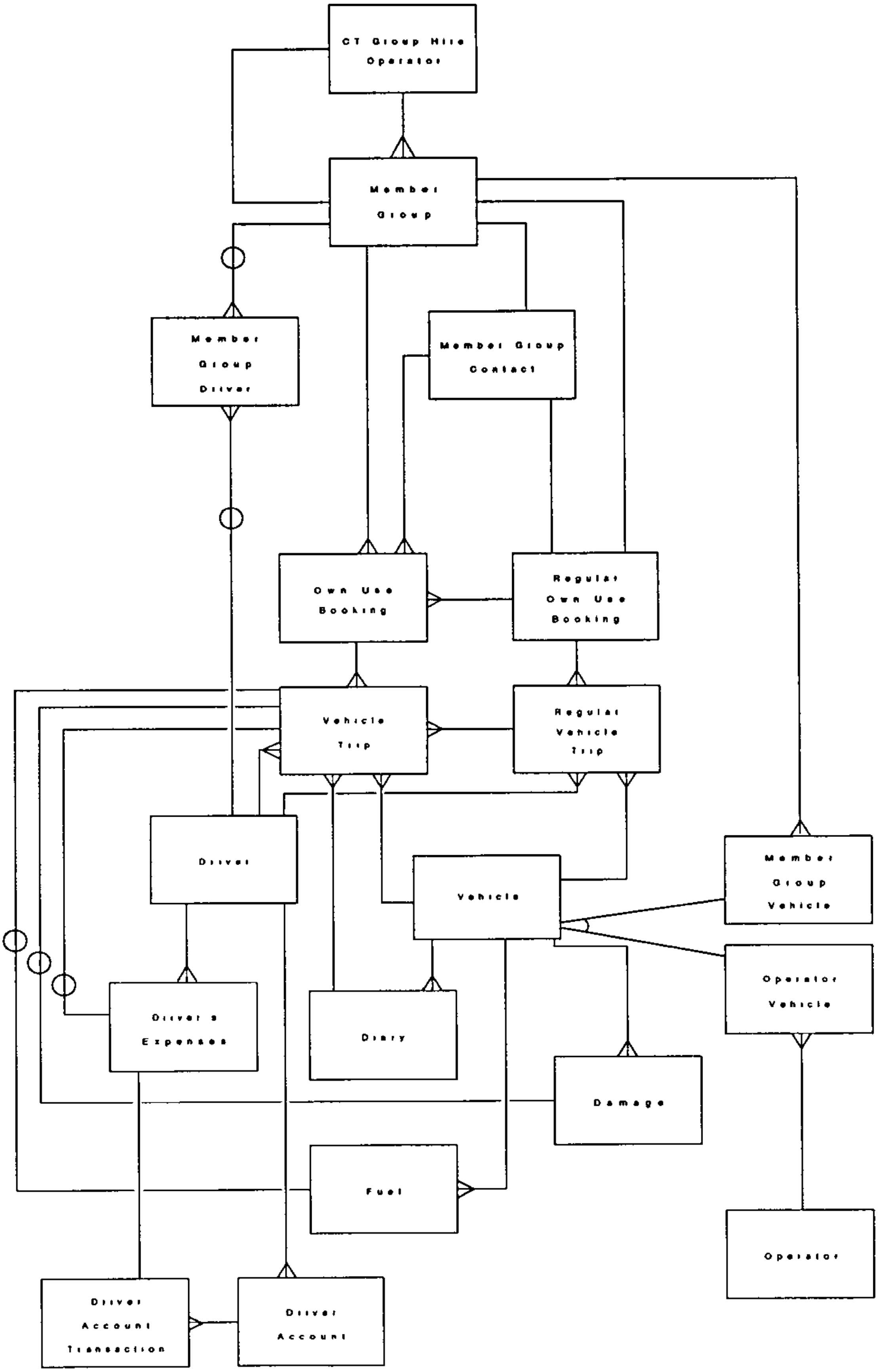


Figure 3.21 *Entity-relationship chart for own use vehicle bookings*

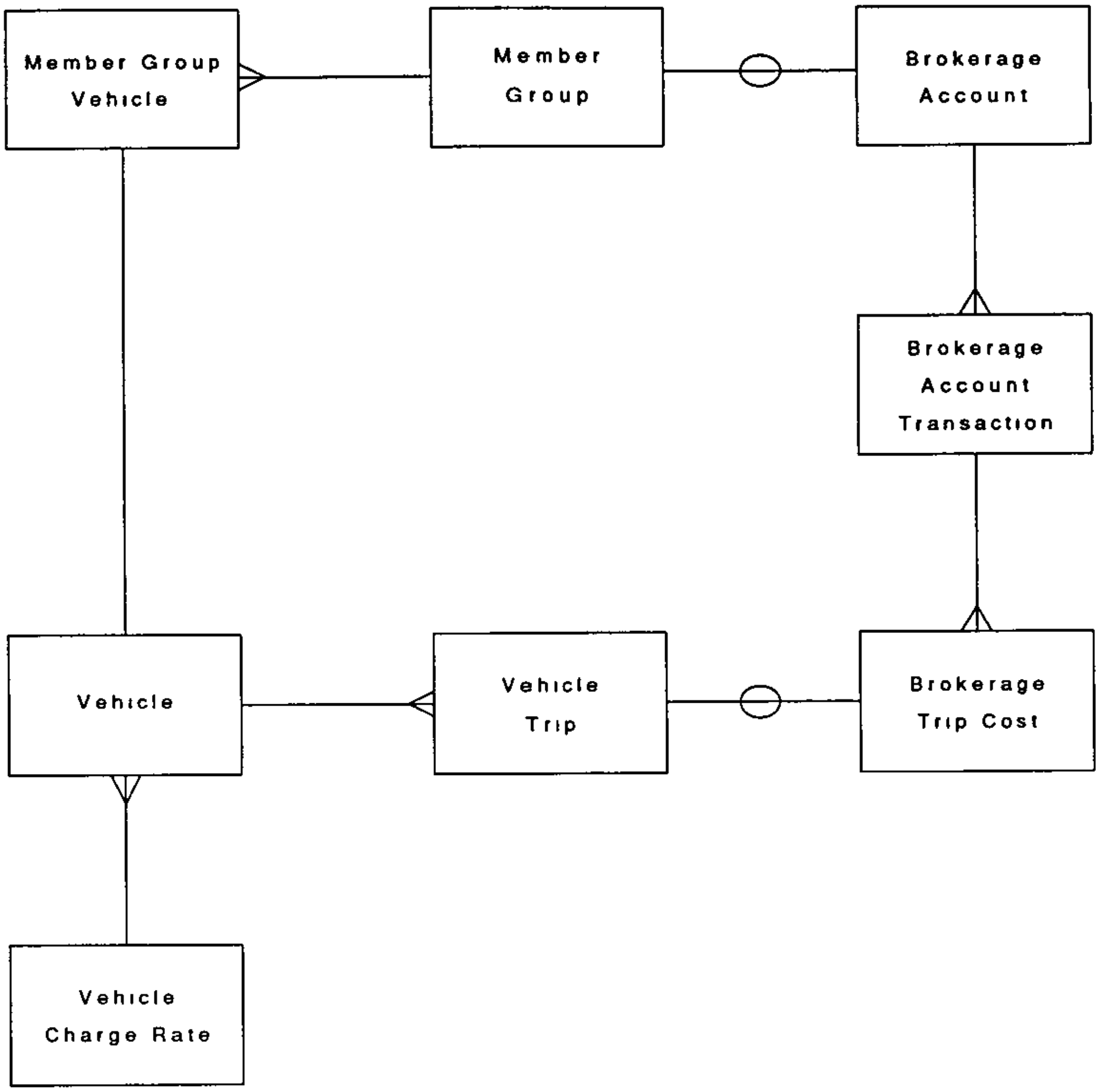


Figure 3 22 *Important entities involved in vehicle brokerage*

which the CT operator would want to send correspondence, the role it plays for example in the Derby CT software. However, analysis shows that there are a wide range of entities in the data model for which it is necessary to hold names and addresses, and experience shows that many of the people, whose names and addresses these are, have more than one role. For example, the secretary of one member group may be a driver for another or the treasurer for a third Similarly, the address of a group may also be the base address for vehicles owned by that group. It was decided at an early stage in the design to use the mailing list to hold all these addresses and to link other entities which use them to the mailing list file. This has two main advantages.

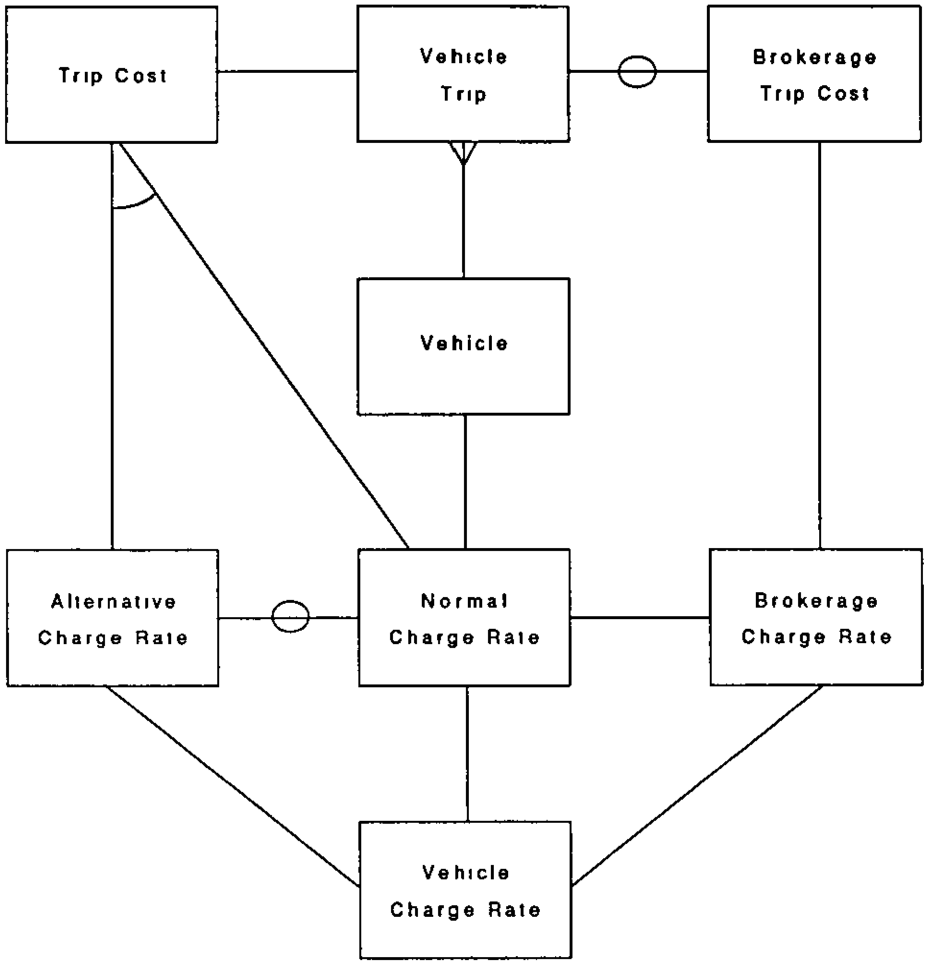


Figure 3.23 *Entity-relationship chart of vehicle charge rates*

- firstly, it reduces the storage requirement on the computer, as any one name and address should only be held once, and
- secondly it eliminates many of the problems associated with updating names and addresses held in several different files when people move.

Table 3 4 shows a list of entities which have relationships to the mailing list file, and the particular types of address which are involved Figure 3.24 shows a summary of the relationships involved for member groups.

Before proceeding to look at the impact of introducing a different type of service, in this case dial-a-ride, into the model, it is worth noting that brokerage affects the processes involved as well as the data model. Figure 3 18 showed the process of taking a group hire booking in the case of an operator where brokerage is not part of the service, based on practice at Nottingham Community Transport. A different

Entitites	Address type
Member group	Group address
	Invoice address
	Contact address
Vehicle	Base address
	Supplier address
	Manufacturer address
Driver	Driver address

Table 3.4 *Entities using the mailing list file for names and addresses*

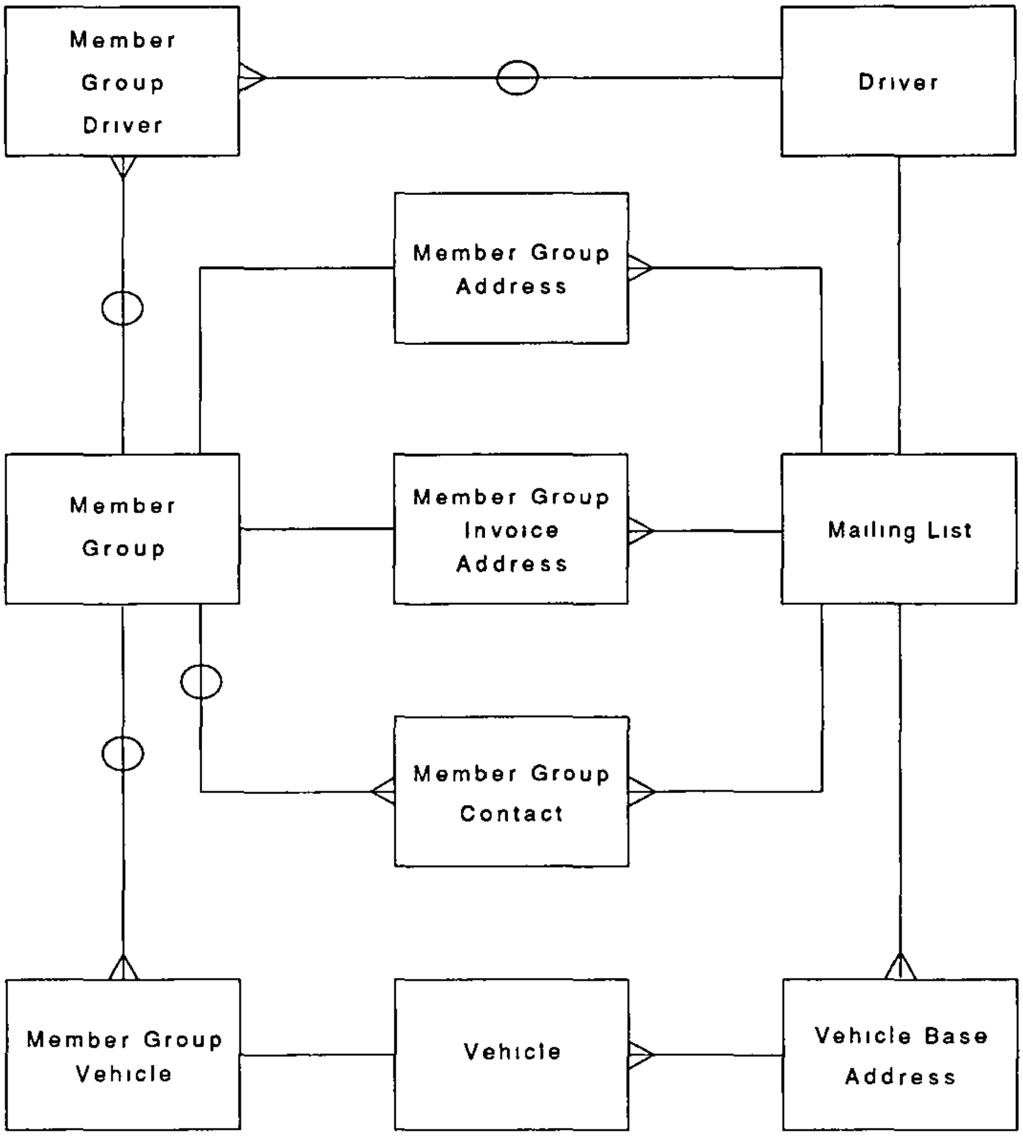


Figure 3.24 *Summary of relationships to the mailing list*

approach is adopted at Birmingham Community Transport which is closely associated with Birmingham Shared Transport Services, a brokerage scheme. In Birmıngham, a booking ıs taken regardless of vehicle or driver availabılity, on the basis that if Birmıngham CT cannot meet the request, Birmingham STS may be able to. A response to a booking ıs not given to a member group straightaway, but only after the brokerage options have been investigated. This ıs shown in Figure 3 25, which may be compared wıth figure 3.18.

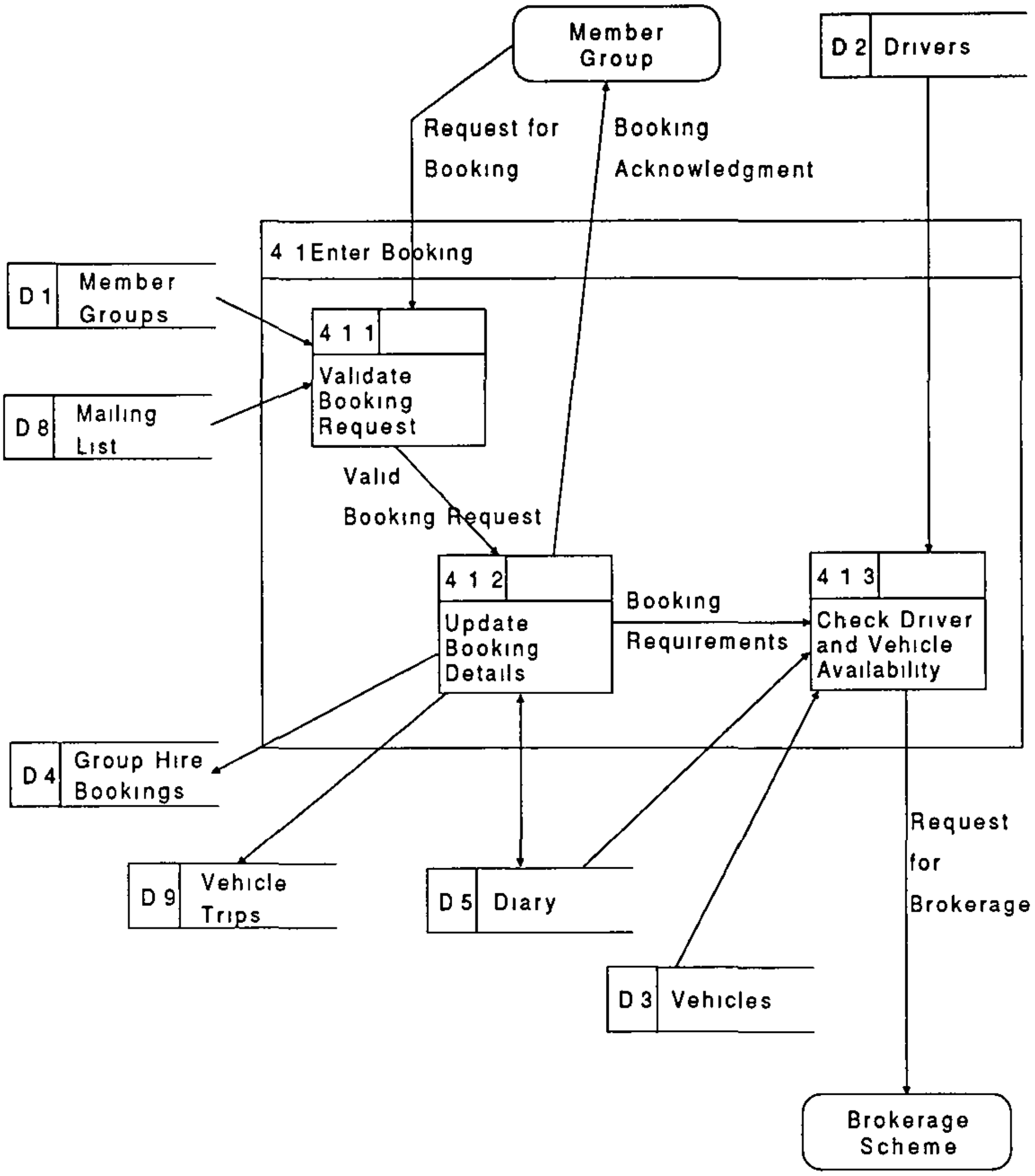


Fıgure 3.25 *Data flow dıagram for Enter Bookıng wıth vehıcle brokerage optıon*

While this functional issue does not directly affect the data model, it is important in the overall design of the system, as booking screens must be designed to enable the user to take both bookings for which the member group is given an immediate response, and bookings which are considered for brokerage. This, in fact, led to the introduction into the design of the idea of allowing a booking to be entered and the vehicle assigned to a trip in that booking being 'To Be Allocated Later' (TBAL). This has implications for the design of the diary which is presented in the next chapter.

Narrative description of dial-a-ride operation

A community transport operator provides a service to individual passengers who through physical disability are transport handicapped, that is unable to use conventional public transport at the level of service which is currently provided.

Passengers are usually required to register in advance with the operator. In some cases, passengers can register and make their first booking at the same time; in others, registration must be completed before a booking can be made Registration is dependent on proof of disability in some dial-a-ride schemes. There may or may not be a membership fee.

The operator operates vehicles which may be owned by the operator or by some other organisation, such as a local authority or public transport authority. The vehicles are adapted with facilities which ease the carriage of passengers with transport handicaps, such as tail-lifts or ramps The vehicles may vary in size and capacity between adapted saloon cars and minibuses. Typically they are no larger than the size of vehicle which may be driven without a PSV licence.

The vehicles are assigned to shifts. During the course of a shift, a vehicle is available to carry one or more passengers and possibly their companions, children or friends on trips within a defined operational area. Whether the vehicle carries one passenger or more than one depends on a mixture of vehicle capacity and the operational policies of the operator.

A shift may be driven by one or more drivers. Drivers may be paid employees or volunteers. Driver's assistants may also travel on the vehicle in order to assist passengers in boarding and alighting from the vehicle.

In order to make a booking, a passenger usually rings the operator in advance. Some operators only accept bookings one or two days in advance, while others accept bookings further ahead, but may operate some kind of rationing system to prevent individual passengers monopolising the service. When a passenger makes a booking, he or she specifies when the trips are required and the origin and destination of each trip. Typically a booking consists of two trips, an outward and a return trip, with the passenger's home address as the origin of the outward and the destination of the return. The despatcher who takes the call enters the details of the booking on a form and then attempts to select suitable vehicle shifts into which to add each trip The shift records may be held on large sheets of paper or on wall boards Passenger trips will be added to shifts with the intention of minimising unproductive vehicle travel, so that passenger trips with origins and destinations close together in space and time will be allocated to the same shift where possible. The despatcher may negotiate times with the passenger in order to enable him or her to travel, or may undertake to renegotiate times with a passenger whose booking has already been accepted. If a booking is refused, a record will probably be maintained for the purpose of shortfall monitoring.

Passengers may require additional help on the vehicle in the form of a driver's assistant, or may require help at their destination. Some Dial-a-Ride schemes provide one or both of these, either through the use of volunteers or through paid employees.

Work sheets are prepared for drivers for each shift, usually a day in advance The worksheets are prepared from a combination of the individual booking forms and the shift records, which usually only contain summary information Despatchers may use this opportunity to reallocate some trips in order to make more efficient use of vehicle resources. This may make it possible to contact passengers whose bookings had been refused and to offer them trips

The worksheets are given to drivers, and may be used to record vehicle odometer readings at each stop in order to calculate distances for statistical purposes. Almost invariably fares are collected by drivers or driver's assistants from passengers. These may be calculated in advance at the time of booking based on a zonal charging scheme or on estimated distance, or may be calculated by the driver based on actual distance. The fares may be recorded on the worksheets or on separate fare sheets.

A driver may record the overall odometer readings and the distance covered by the vehicle during the shift on the worksheet for that shift or as a line of detail on a vehicle log sheet.

In the cases where the driver has not collected the fare from a passenger, an invoice must be prepared from the details on the worksheets. This can happen in cases where the passenger is unable to deal with money through disability or where the trip or trips are being paid for by some other individual or agency such as a social services department.

In most cases, travel is subsidised in some way, and returns must be prepared from the information on the work sheets for submission to the subsidising authority.

Analysis and integration of entities involved in dial-a-ride

Reading through the description above, a list of entities involved in the operation of a dial-a-ride service can be drawn up, and this is listed in Table 3.5.

It was recognised that many of these entities had already been identified as part of the analysis of other services. However, their involvement in dial-a-ride may require them to be treated in a different way throughout the system Other entities are specific to dial-a-ride, and it is with these in particular that this section is concerned. These are listed in Table 3 6

Entities
passenger
companion/child/friend
operator
vehicle owner
vehicle
booking
vehicle shift
driver
driver's assistant
passenger trip
refusal
volunteer help
work sheet
fare sheet
vehicle logsheet
invoice
source of payment

Table 3.5 *Entities involved in dial-a-ride*

Entities

passenger
companion/child/friend
booking
vehicle shift
driver's assistant
passenger trip
refusal
volunteer help
work sheet
fare sheet
invoice
source of payment

Table 3.6 *Entities specific to dial-a-ride*

The most significant of the additions is the individual passenger. While group hire and brokerage are concerned with the transport of groups of people, who may be anonymous to the operator, dial-a-ride is concerned with the transport of named individuals, about whom records must be maintained. A passenger may make a booking which consists of one or more passenger trips. A passenger trip is very different from the vehicle trip entity which was identified as part of the group hire analysis. Only one vehicle trip can take place on a particular vehicle at any one time. However, many passenger trips may take place on the same vehicle, and may overlap in time. The vehicle shift in dial-a-ride is coterminous with the vehicle trip, and a number of passenger trips may be related to a single vehicle shift. This relationship is shown in Figure 3.26

The vehicle shift, however, involves more than the simple vehicle trip. More than one driver may be associated with a vehicle shift in some dial-a-rides, although many make driver shifts and vehicle shifts coterminous Driver's assistants may also be associated with vehicle shifts. A volunteer helper or escort, on the other hand, is associated with the passenger trip, as escorts usually meet passengers at their destinations in order to assist them The same individual may work in either a voluntary or a paid capacity as an escort or driver's assistant, and records need to be maintained about such people. Such individuals are part of the dial-a-ride operation, and are not to be confused with the individuals who are called escorts in some dial-a-rides, who are friends or relatives of the passengers and who assist them to travel, and who may be charged a reduced rate for travel While it is necessary to

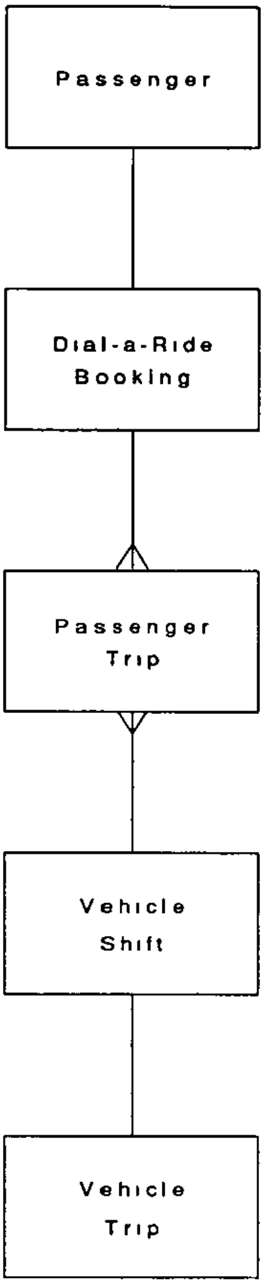


Figure 3 26 *Dial-a-ride passenger booking entities*

maintain records of the numbers of such people for the purposes of allocating space on vehicles, along with companions, friends and children travelling with passengers, they are not the subject of permanent records in the system.

Worksheets are used in dial-a-ride to provide drivers with details of the passenger trips involved in a shift They are not a separate entity in the system, but simply a document which contains information about a vehicle shift. Similarly fare sheets, which may sımply be a part of the worksheet which is completed by the driver, are documents whıch may result in further data being entered into the system after a shift has taken place. Figure 3.27 shows an elaboration of Figure 3.26, involving these addıtional entities.

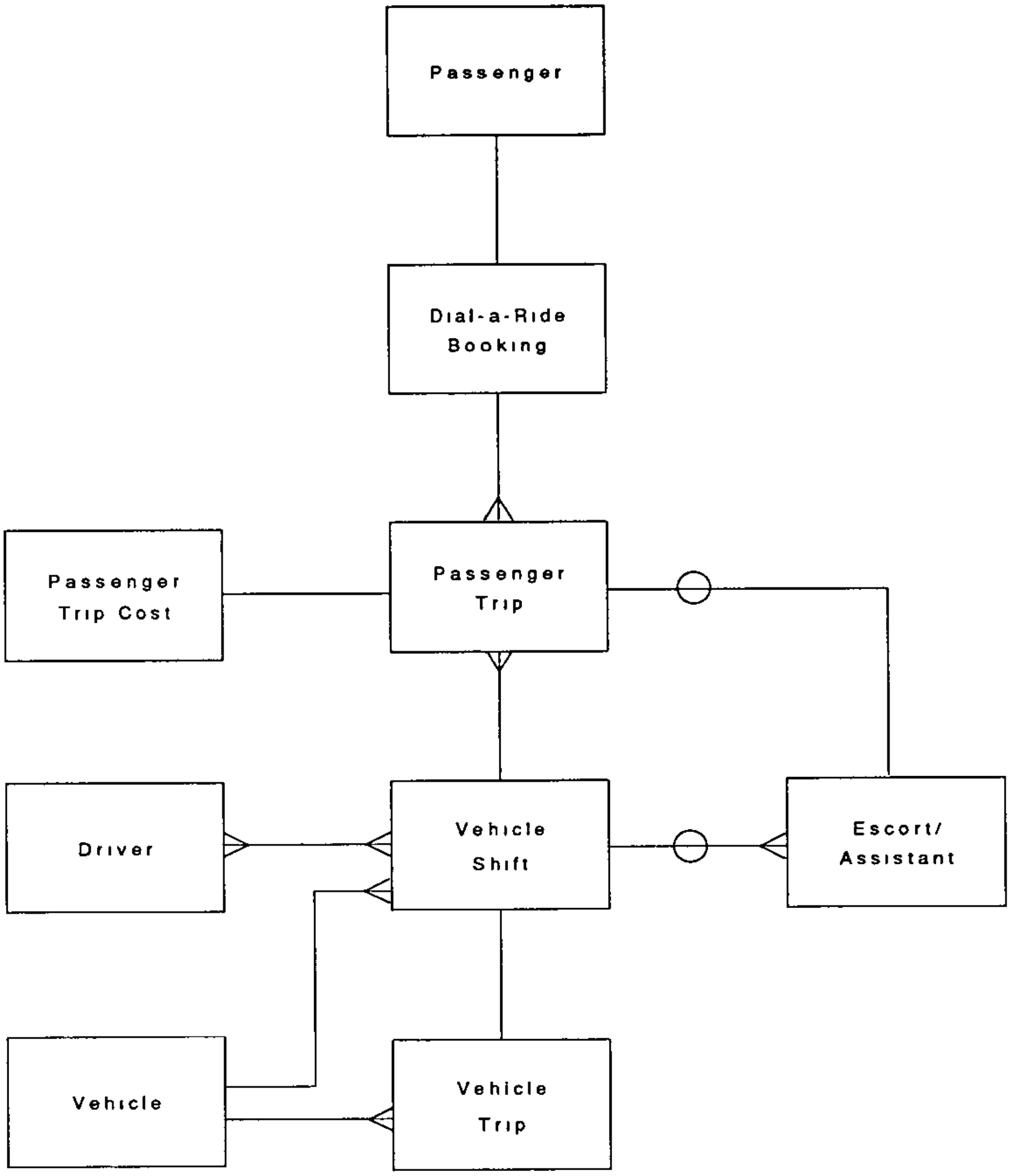


Figure 3 27 *Elaboration of dıal-a-ride passenger booking entıtıes*

However, fares are usually calculated in advance for passengers and are based either on a zoned system or on estimated dıstances between map references for the start and finish of each passenger trip. As part of the dial-a-ride system, it is necessary to maintain records which support the different methods of charging. There will be further discussion of methods of calculatıng distance for the purpose of scheduling ın Chapter 4. For the moment it is necessary to analyse the possible methods of chargıng. The dial-a-ride operators analysed during the research all used a standard charging rate as the basis for their fare calculations; exceptions to the standard rate are based on whether the passenger uses a particular kind of bus pass. Thus, as in group hıre, a charge rate may have alternatives. The charge rate itself may be based on the estımated, calculated distance between map reference areas, or may be based on set rates between zones. Fıgures 3.28 (a) and 3 28 (b) show the applıcatıon of the two methods to a passenger trip from source to destination. In the fırst case, the dıstance can be calculated using Pythagoras' method, and the fare for that dıstance calculated. In the second case, there ıs a fixed fare for travel between the two zones, which is applıed to the example. Fıgure 3.29 shows the entitıes involved, and ıncludes some of the entıtıes shown in Figure 3.27.

Fınally, it ıs possıble that some passengers wıll not themselves pay for each trıp when they make it, eıther because they do not carry money or because the trip ıs to be paıd for by some other agency, such as a social services department or an old people's home. It is therefore necessary to be able to ınvoice for trips, and for these invoices to be allocated to accounts whıch may be paid eıther by the passengers themselves or by other agencıes This is shown ın Fıgure 3 30. It is ımportant to note that one account can be used by a number of dıfferent passengers, for example, an account used by an old people's home.

Identification of common entities

The aim of the software development was not sımply to provide packages for specific types of CT operation, but to provide a package which ıs both modular and ıntegrated. The modularıty of the software is catered for in a straightforward manner, by the devlopment of a separate module for each mode of operation, and the analysıs descnbed above outlınes this process for group hire, and dıal-a-ride The development of modules for social car schemes and for dial-a-bus has been

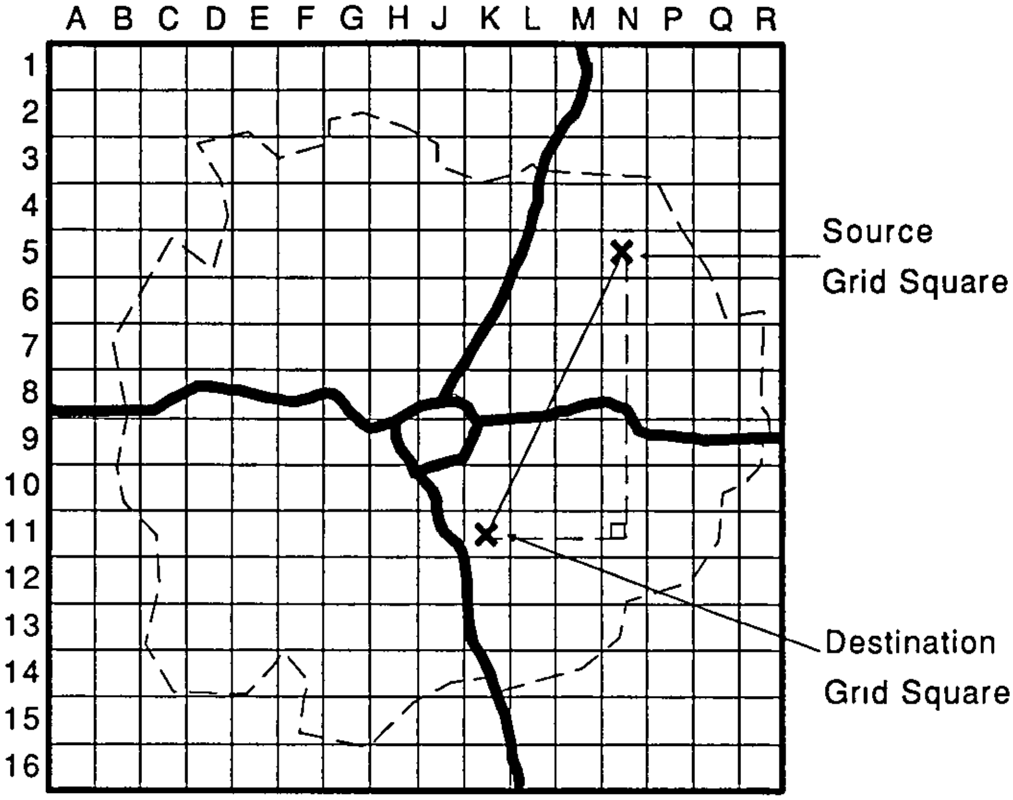


Figure 3 28(a) *Map reference based calculation of fares*

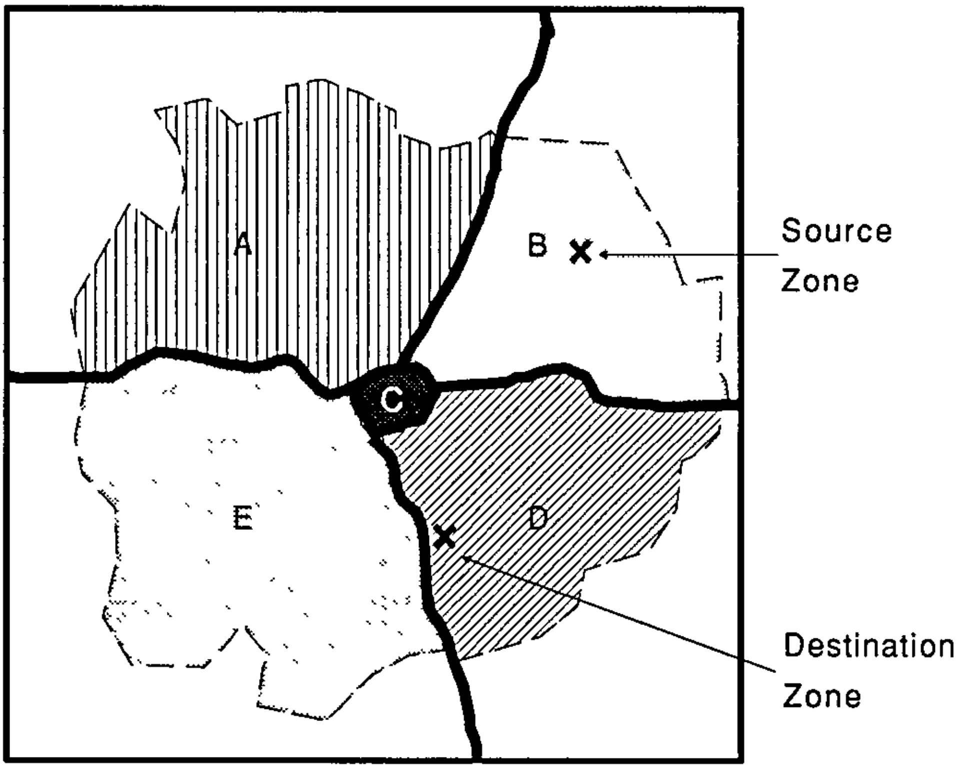


Figure 3.28(b) *Zone to zone based calculation of fares*

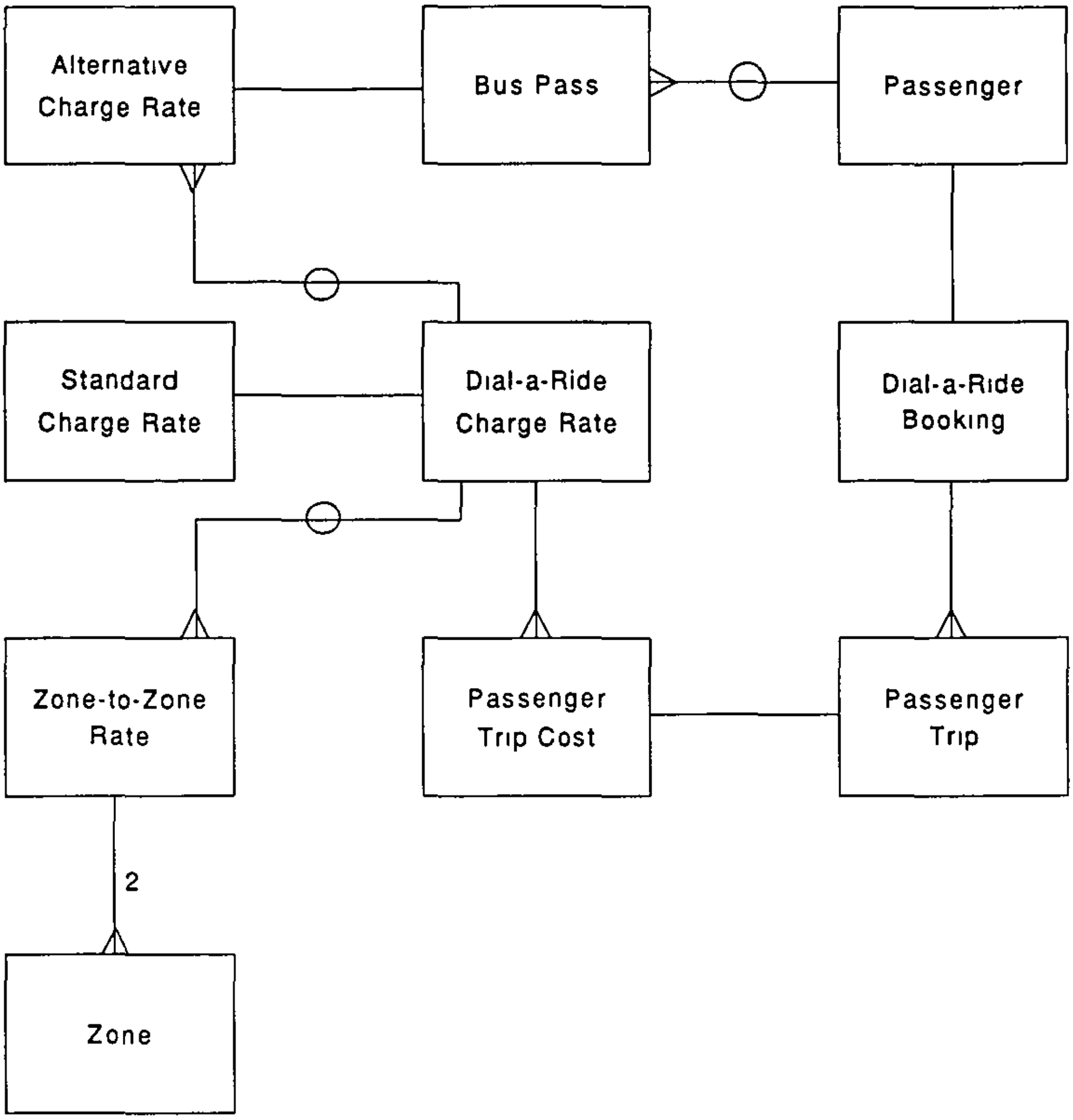


Figure 3.29 *Dial-a-ride charging entities*

continued in the same way. This is common practice in software development and may be compared to providing an accounts package as separate modules for sales, purchasing, payroll and so on.

The issue of integration is less straightforward, and is addressed by this section of Chapter 3 It was noted in the first part of this chapter that none of the packages analysed there, apart from the Derby CT software, which is not concerned with bookings, makes provision for more than one type of CT service. In reality, many CT operators run more than one type of service, as has been described in Chapter 1. It is, therefore, important that each module should work in conjunction with every

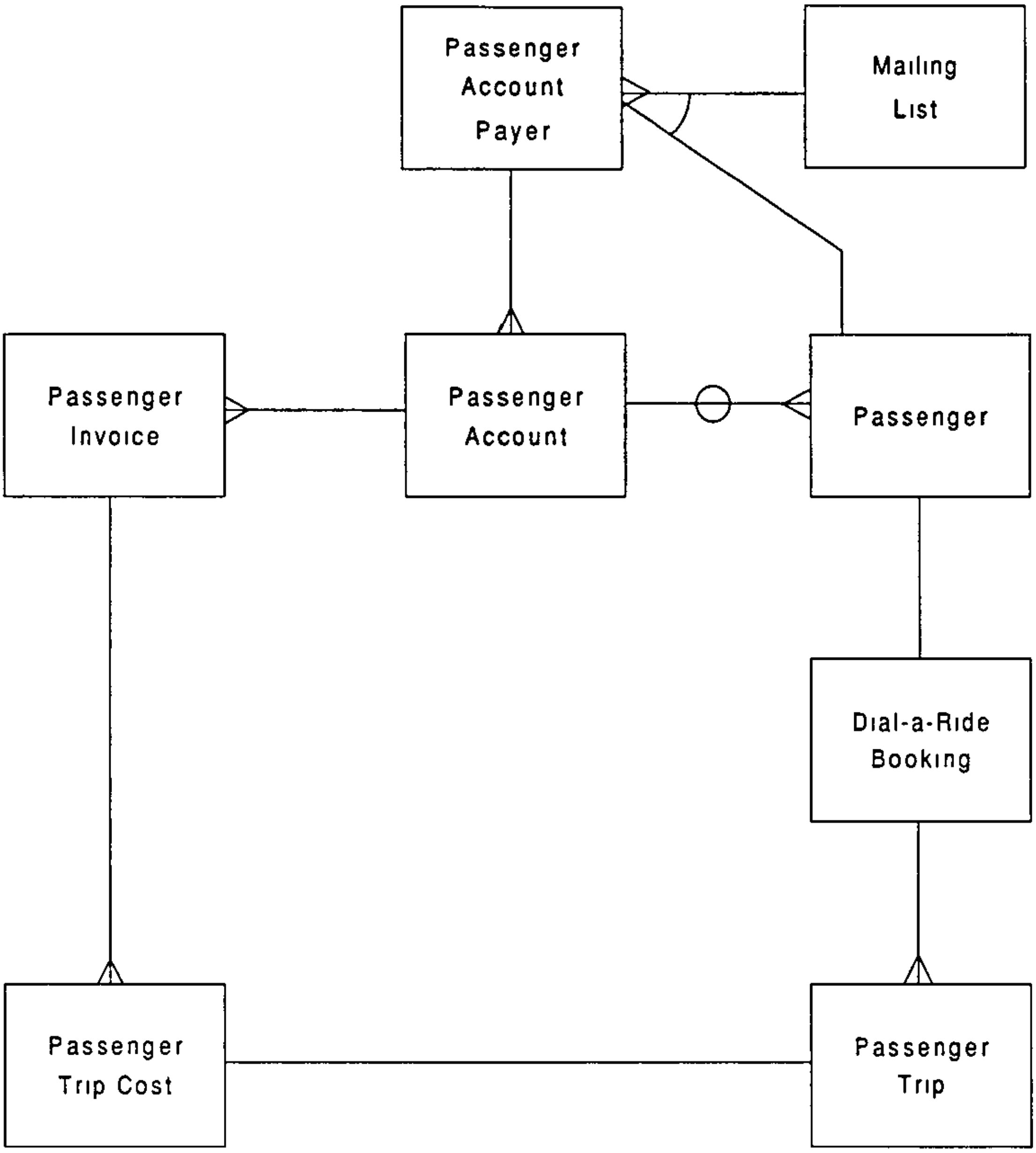


Figure 3.30 *Passenger accounts*

other module, and that there should be no overlap of function or data storage among the different modules. In Tables 3.5 and 3 6 above, it was noted that some of the entities identified for the dial-a-ride module had already been identified in the context of group hire. For both the group hire module and the dial-a-ride module to work, a vehicle entity, which will be represented by a file in the finished software, is required

There are three possible approaches to this:

1. Provide a vehicle file in each module, one for group hire vehicles and one for dial-a-ride vehicles. The problem with this approach is that vehicles may be used by both services, and we are then obliged to update two separate records in different files. Furthermore, as new modules are added, new vehicles files are also added.

2. It therefore makes sense to locate all vehicle records in a single file. This raises the question of which module this file should be located in. If a user only purchases one module, then the vehicles file must be provided, but if they subsequently purchase a second module for another service, then a duplicate file should not be introduced.

3. It is therefore necessary to take the vehicles file out of the service modules and make it available to all of them.

The same principle applies to a number of other entities which have been identified above, such as drivers, mailing list and vehicle trips. A number of entities can be identified which are common to more than one module, and it was therefore decided to provide these in a separate Common Module. This decision was made at an early stage in the analysis of the system, based partly on the recognition of the weaknesses in some of the other available packages and partly on the recognition of the factors described here To compare the design with that of an accounts package again, this is comparable in some ways to the provision of a separate nominal ledger module which integrates with the sales, purchasing and payroll modules However, it goes beyond that, and may be compared with the recognition that both sales and purchasing may require a file of VAT codes and rates, and that this is best provided by a single file rather than by separate files in each module.

A list of common entities can be derived from the lists and the entity-relationship charts used above. The most important of these are shown in Table 3.7.

The structure of the system is shown in Figure 3.31. Each of the modules which provides for a specific type of CT service makes use of data entities in a common module. Because of the nature of the PICK operating system and the System Builder applications generator, the data dictionary definitions and the screen definitions are also held in the dictionary level of the data files that represent these entities.

Entities
driver
driver's expenses
driver accounts
driver account transactions
driver's assistants
fuel
mailing list
own use bookings
own use regular bookings
passengers
passenger accounts
passenger account transactions
vehicle
vehicle operators
vehicle trip
vehicle damage

Table 3.7 *Common entities*

Most of the files listed in Table 3 7 represent static entities, such as vehicles and drivers, which do not change much over time. However, the purpose of the system is to allow bookings for the various services to be maintained, and the vehicle and passenger bookings, the shifts and the vehicle trips represent the dynamic data which is held in the system and which is changing from day to day. Of the common

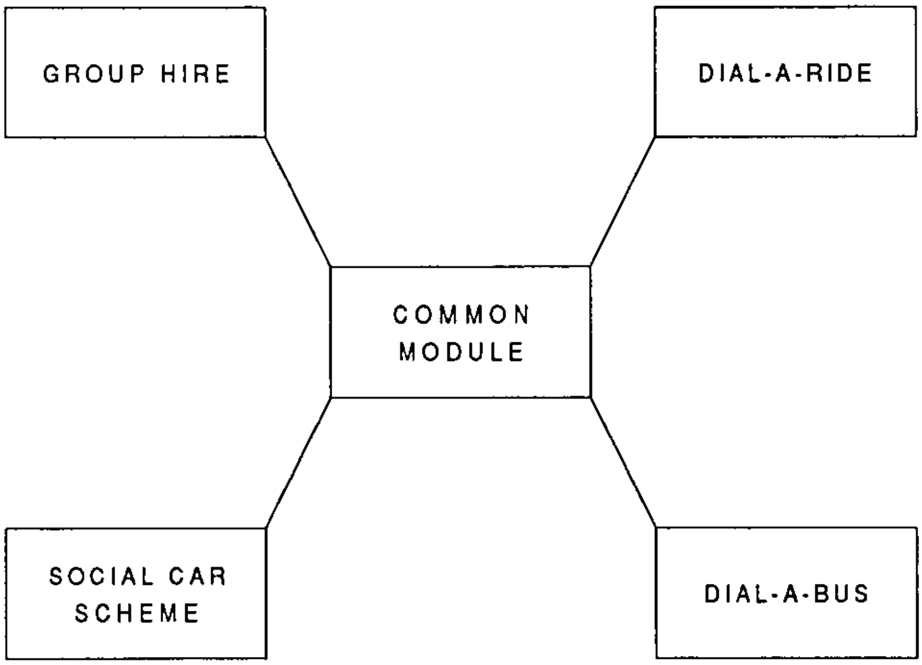


Figure 3.31 *Modular software structure*

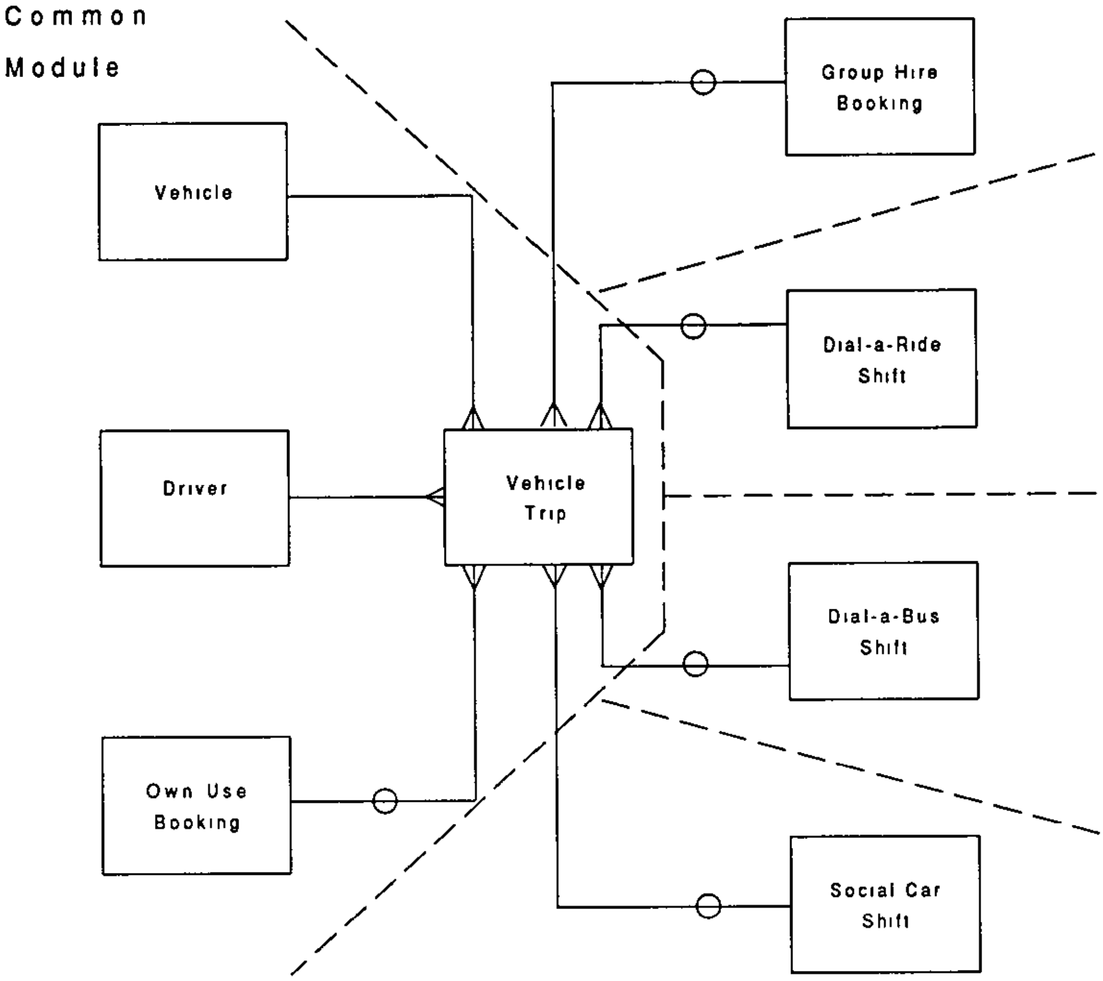


Figure 3 32 *Relationships to vehicle trips*

entities, it is the vehicle trip entity which is central to this aspect of the system. Any use of any vehicle for any service is represented by an instance of the vehicle trip entity. As was discussed above in the section on the dial-a-ride analysis, only one vehicle trip can take place on any one vehicle at any time However, a vehicle trip could represent a trip which is part of a group hire booking or a dial-a-ride shift, or use of the vehicle by the vehicle owner, or with the addition of other modules, a social car scheme trip or a dial-a-bus service. This is shown in Figure 3.32 The boundaries between the various modules are shown by dotted lines.

As it stands, there is no way within the data structure outlined here to maintain the constraint that vehicle trips may not overlap with one another This feature of the system is explained in Chapter 4.

This chapter has explored the data structures of four software packages for CT operators, and outlined some of their weaknesses. In the light of these data structures and based on the analysis which was carried out during the research, the data requirements for group hire and dial-a-ride operations have been identified. Based on these two types of service, the existence of entities common to both was identified, and the need for a separate common module established.

Many of the operations required by users are simply operations to maintain this database: to create, update and delete records which represent instances of the entities which have been identified by the data analysis However, certain operations are required which mean that users require more than just a database management system. The ability rapidly to check the availability of a vehicle for a trip, to maintain the constraint that vehicle trips do not overlap, to allocate passengers to dial-a-ride shifts, to find the most suitable brokerage vehicle for a booking, all these aspects of the requirements need more complex data structures and algorithms to manipulate the data These elements of the system are explored in Chapter 4.

Chapter 4

Diary, Brokerage and Scheduling Features

Introductory Points

Chapter 3 covered the analysis of the basic data structures required for CT operations. This chapter examines the way in which four operational functions have been designed and implemented in the software. Within the operation of a CT, as in any business or other organisation, there are a number of different types of processes which are performed by people, whether they are working with a manual information system or a computerised one·

- storing information;
- retrieving information,
- producing reports; and
- making decisions

Storing information may involve recording details of passengers on index cards or entering a group's booking on a booking form in a manual system, or the entering of passenger details or a booking into a database Retrieving information may involve looking up passenger details on an index card or finding a booking sheet in a filing cabinet, while in a computerised system, index files are used to assist users to retrieve passenger details by surname or booking records by the name of the group which made the booking. In a manual system, producing reports can often be a tedious business, involving the extraction of data from large numbers of written records, for example to analyse the passengers by the areas in which they live, whereas in a computerised system, if the information requirements of the organisation have been properly defined, and the necessary data has been stored in the system, producing regular or ad-hoc reports can be a relatively simple task and one which does not involve considerable expenditure of clerical effort. Finally decisions are made at all levels within the organisation, from those about day-to-day operational matters such as the allocation of passengers to dial-a-ride shifts or the availability of vehicles for bookings, to longer term strategic decisions about service development made by management committees. The four functions examined in this chapter fall into the category of operational decision making. In each case, an understanding of the ways in which people make the decisions is necessary in order to design computerised approaches to those decisions.

The first part of this chapter explores aspects of how people make decisions and solve problems, with reference to aspects of management theory and cognitive psychology. It then proceeds to examine how computers can be used to assist in this process, with particular reference to the concept of decision support systems. Finally, each of the four functions is discussed in terms of how people perform that function, the range of computerised approaches which are available, where appropriate, and the approach which has been adopted in the software.

At this point, it should be remarked that the emphasis has been on developing approaches which support or assist the human user in making decisions rather than on those which replace the human decision maker with a purely computer-based or mathematical solution. This stems from the community development approach of CT itself and of the project, which is concerned with the development of people's skills rather than with deskilling or the replacement of people with machines.

How People Make Decisions

Some authors regard problem solving as a stage in decision making, while others regard decision making as a stage in problem solving, and yet others seem to regard the two terms as effectively synonymous.

Some authors, such as Turban (1990), seem to use the terms interchangeably. Authors on management decision making such as Cooke and Slack (1991) regard decision making as a stage (or series of stages) in a larger process of problem solving. While some cognitive scientists regard decision making as a minor step in the process of solving some human problems (Mayer, 1983; Johnson-Laird, 1988). This latter view seems to be based on a view of decision making which limits it to the choice between options or to the yes/no decision, rather than seeing it as a process.

Decision Making

Models of human decision making are largely rooted in study of decision making in business and organisational contexts. A general model of the processes by which humans reach decisions has been put forward by Simon (1960). He proposed three stages of the decision making process:

- Intelligence
- Design
- Choice

Other authors have added either Implementation or Review or both to this list. Thus the total model can be seen as:

- Intelligence
- Design
- Choice
- Implementation
- Review

and the introduction of Review implies a cycle, with the original decision subject to evaluation and the possibility of repeating the process.

The Intelligence phase involves an alertness to situations, whether problems or opportunities which require a decision, and the gathering of data about the situation. Design involves the generation of alternative courses of action and their analysis, and the generation of criteria by which the decision is to be made. The Choice phase is what some authors see as decision making, that is the choice between alternatives using the criteria determined during the design phase. Implementation is concerned with putting the outcome of the decision into practice, and is clearly critical in the context of strategic business decisions The Review phase involves an evaluation of the decision and its impact to ensure that it has been implemented and that it has had the desired effect. This is not seen as a purely linear process, it is possible to repeat phases, as is shown in Figure 4.1.

Cooke and Slack suggest a more complex model of the decision making or

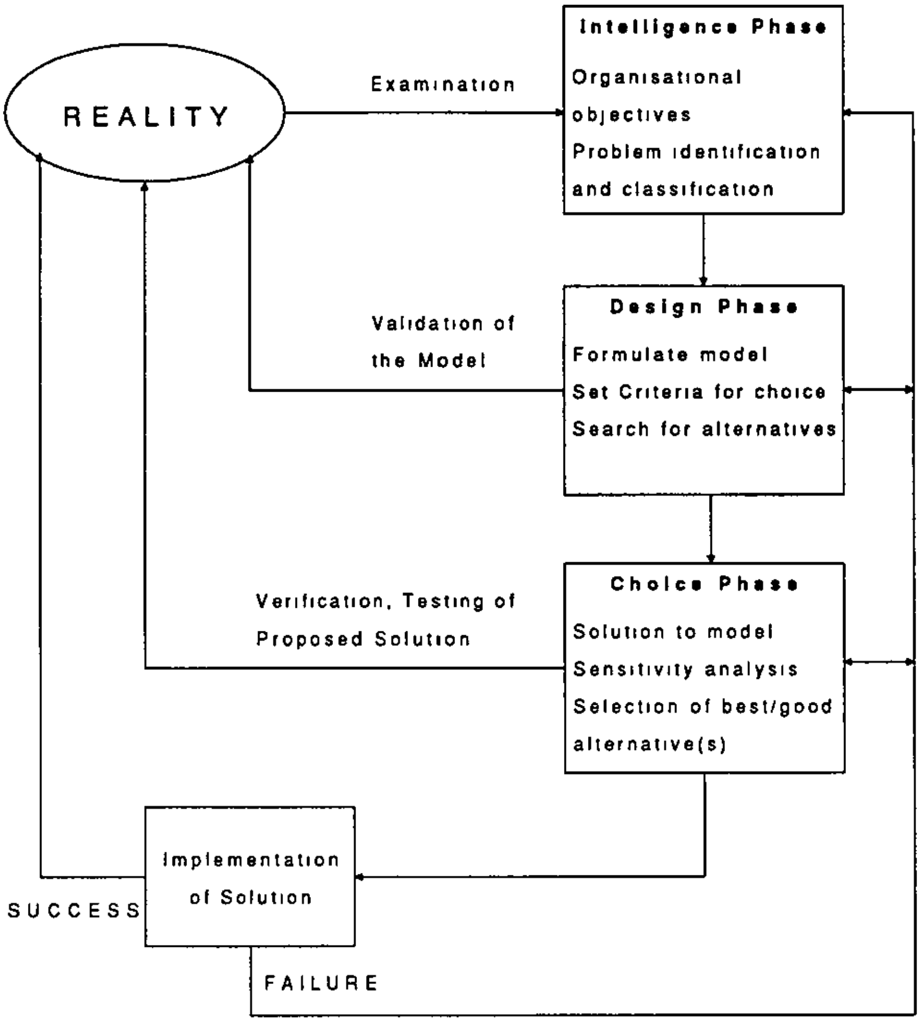


Figure 4.1 *Simon's model of decision making (after Turban, 1990)*

problem solving process, which they view as a cycle (Figure 4.2) This has the following phases:

- Observe/monitor
- Recognise problem
- Define problem
- Understand problem
- Determine options
- Evaluate options
- Choice
- Implement

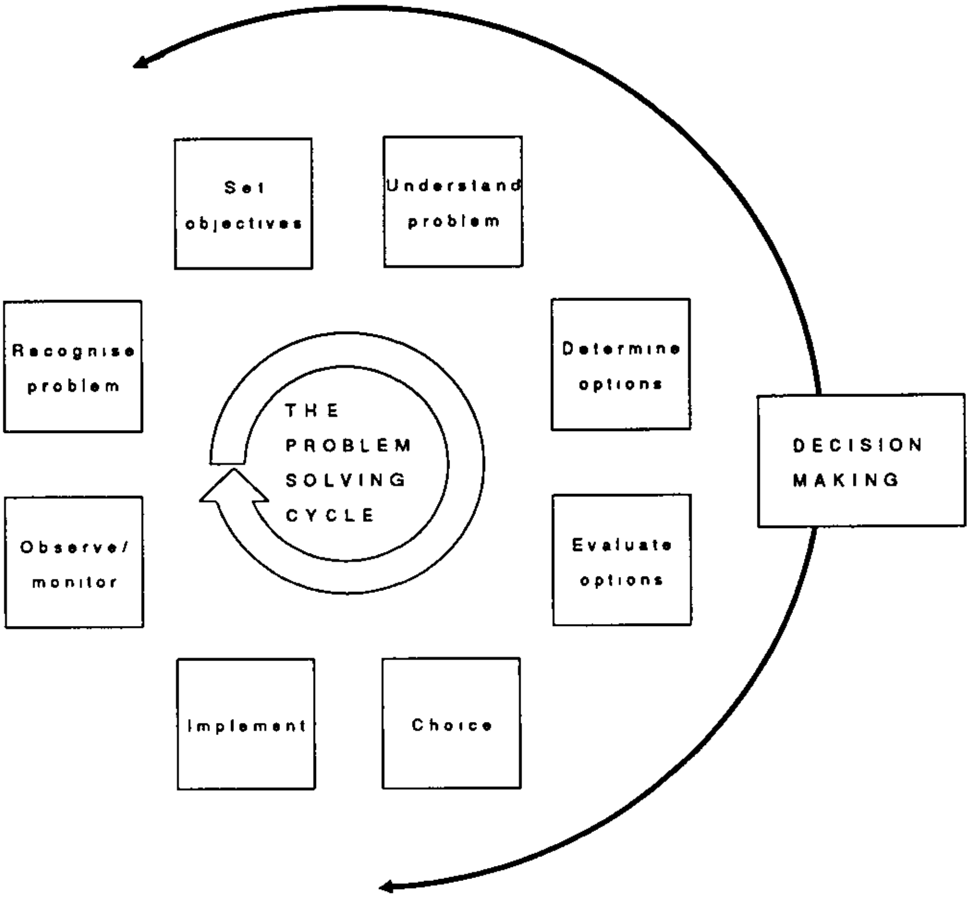


Fıgure 4 2 *Decısıon making as part of Cooke's and Slack's problem solvıng cycle*

Only five of these stages are part of what they regard as the decision makıng part of problem solvıng. Again, the process can involve repeatıng stages.

Two classıfıcatıons of decısion problems whıch have been proposed by theorists in the field of business decısıon making are worth highlıghting here The first concerns the level of management decisıon makıng and assocıates decisions wıth the levels of the well known triangular model of the levels of management within an organisation (Anthony, 1965) as in Fıgure 4 3. Decısıons may be strategıc, tactical or operational. Strategic decısıons are concerned wıth long term plannıng and wıth corporate objectıves, such as product development or investment strategy. Tactical decısions are made by mıddle management and are concerned wıth the achievement of medıum term targets Operatıonal decısıons are concerned with the day to day activities of the organısatıon. In the context of CT, strategıc decısions mıght ınclude

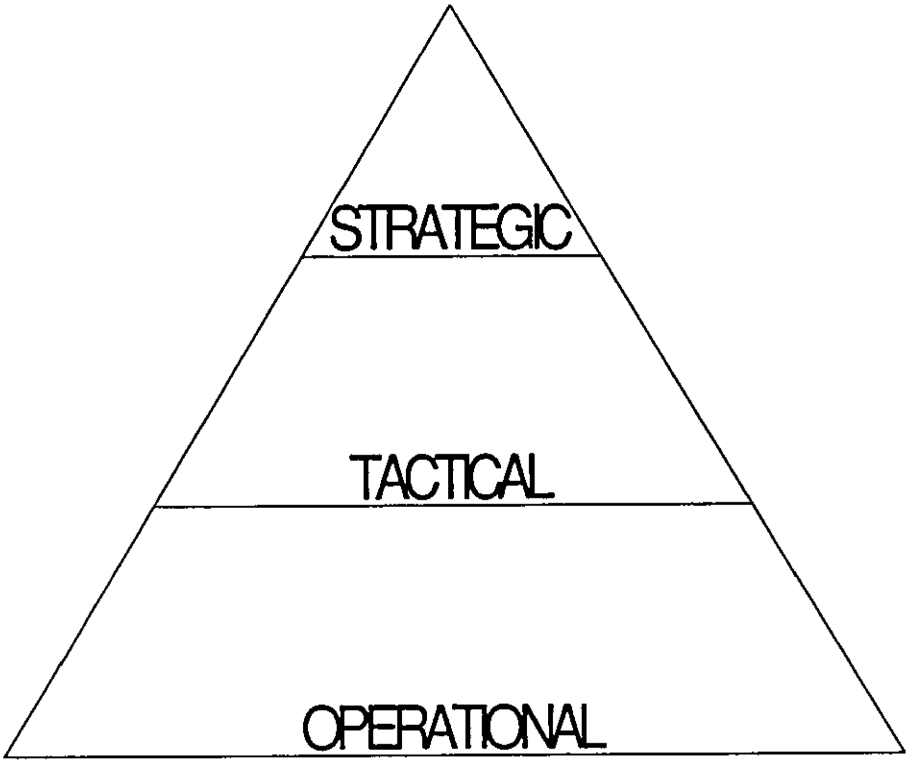


Figure 4.3 *Anthony's model of levels of decision making within an organisation*

whether to tender for a new service, whether to introduce a dial-a-bus service in order to try to increase vehicle utilisation compared with a dial-a-ride service, or when to replace vehicles; tactical decisions might include what vehicle model or conversion to purchase, or how to attract additional volunteer drivers; operational decisions include the many decisions taken by despatchers, booking clerks and volunteer coordinators who allocate the resources of a CT to meet the demands of its users.

The second classification concerns the nature of the problem or decision which is to tackled. Various authors have proposed different terms for the two extremes of this dimension. Simon (1960) suggested that problems are either programmed or nonprogrammed. Programmed problems are those for which standard, structured solutions exist, and these are typically routine and repetitive processes; nonprogrammed problems are those complex, 'fuzzy' problems for which no standard solution exists. The terms structured/unstructured and clerical/intuitive have also been used to define this dimension. Gorry and Scott-Morton (1971) introduced the idea of semi-structured problems and decision processes, based on Simon's work.

Semi-structured problems are those for which some of the phases of the decision making process are structured and some are unstructured. For example, the development of a new product may be a semi-structured problem: the identification of a gap in the market and the creative design of a new product represent unstructured intelligence and design phases, while the evaluation of the new product using market research techniques and financial modelling is a process which is well understood and structured.

There is a tendency for these two dimensions of problems to be related. Typically, strategic problems are less structured and less routine, they require more intuitive thought and creativity, while operational problems are likely to be more structured and routine. This is not to say that all strategic problems are nonprogrammed - investment decisions can be carried out using well understood, programmable processes - or that all operational problems are clerical - matching volunteers with people who need help requires consideration of more than just availability, and may be largely intuitive.

These aspects of decision making and problem solving will be considered again in the context of how computers are used by people to solve problems, and specifically in the context of the class of computerised systems known as decision support systems. Before that, two other models of how people solve problems are worth examining.

Problem Solving

Newell and Simon (1972) proposed a model of human problem solving based on an analogy between computer processing and human information processing. This model, shown in Figure 4 4, consists of a number of elements or subsystems These are:

- Perceptual subsystem
- Cognitive subsystem
- Long-term and external memories
- Motor subsystem

Each of these performs a specific function in the processing of information. It

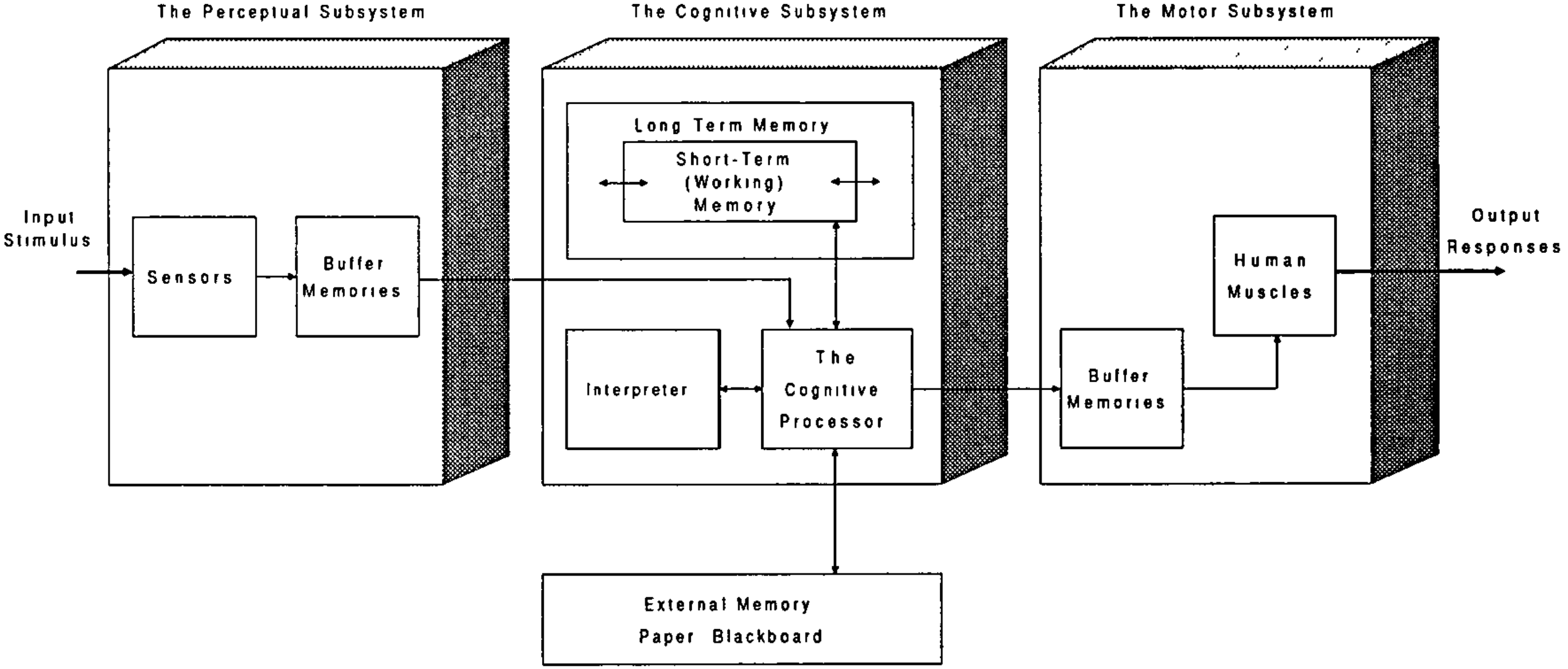


Figure 4 4 *Newell's and Simon's model of human information processing* should be stressed that this model is a high level model of human cognition, rather then one which attempts to explain cognition at the level of neurons and brain activity.

The perceptual subsystem handles the input of external stimuli through sensors such as the eyes and ears. As well as the sensors, the perceptual subsystem contains buffer memories which hold information on a short term basis before it is passed on to the cognitive subsystem.

The cognitive subsystem takes information from the sensory buffer memories and processes it in conjunction with information which is retrieved from short-term or working memory or from long-term memory. In Newell's and Simon's model, the cognitive subsystem consists of three modules, the cognitive processor, which is equivalent to the central processor in a computer system, the short-term or working memory and the interpreter which interprets the 'programs' required in order to process information. Different kinds of cognitive tasks require different amounts of processing. Knowledge may be based on facts, algorithms or heuristics. For example, if someone is asked what 8 times 4 is, she will retrieve it as a fact, asked to multiply 127 by 276, she will have to apply an algorithm in order to perform long multiplication; asked what 127 times 276 is approximately, she will use a heuristic to estimate that 127 x 276 is roughly the same as 100 x 300, and therefore in the region of 30,000. For some tasks, the information required will be available from sensory inputs and working memory, for others it will be necessary to retrieve it from long-term memory

Long-term memory is a system which stores large numbers of symbols with a complex indexing system There are a number of hypotheses about how information is stored in long-term memory; a useful view is that the fundamental symbols of memory are combined into 'chunks' which represent units of stored information, digits, symbols or words associated with a pattern of stimuli. Information is stored in a hierarchy of chunks, and the recall time for a chunk is only a few tenths of a second, but the time required to commit information to memory is between 5N and 10N seconds, where N is the number of symbols involved.

An important feature of memory is its storage capacity. While the storage capacity of long-term memory is essentially unlimited, working memory can only store 'the magical number seven, plus or minus two' chunks of ınformation (Miller, 1956), and the usual estimated duration of short-term memory ıs 20 seconds. Moreover, it is believed that only about two chunks can be retained in working memory while another task is being performed, that is if the person is interrupted. Humans typically support the decision making or problem solvıng process through the use of external memories, such as notes on paper or blackboards.

The lımitations of short-term memory may be partially overcome by the use of alternative representations of information such as analogıes or graphical displays. The use of a graph to represent informatıon rather than a table of figures may requıre fewer chunks of storage ın order to hold the essentials in short-term memory.

Fınally, the motor subsystem receıves ınformatıon from the cognitıve processor, which results in the inıtıation of activity in other parts of the human system, and observable behaviours such as moving or talkıng.

To descrıbe how humans process ınformation using this model of the cognitıve system, researchers ın the fıeld of artificial ıntellıgence have devised a kind of programmıng language called a production system, whıch consists of 'ıf-then' statements called productıon rules, and the workıng memory External stımuli are perceıved by the sensory subsystem and transferred into working memory, where they trıgger the if clauses of ıf-then rules The then clauses specıfy appropriate actions and are converted ınto responses.

A further theory of problem solvıng which wıll be of use here is the concept of problem solvıng as a search through a problem space Wickelgren (1974) defines a problem as having three elements.

- givens,
- operatıons,
- a goal.

Gıvens represent the inıtıal state of the problem situation, and include information about objects, facts, assumptions, and axioms. Operatıons are the actions which may be carried out on the elements of the problem in order to transform the sıtuatıon from the inıtıal state towards the goal state. The goal represents the desıred state towards whıch the problem solver is working Thıs model can be simplified into states and operations, in the sense that the initial situation and the goal are both instances of different states of the problem situation. The possible states can be represented as a tree or network, with operations forming the links between nodes or states. The network represents the problem space of the of the problem, containing all the possible states and paths between them. Problem solving is then a search through the problem space in order to find one or more goal states. This model has its roots in mathematical problem solving, but will be useful in considering human problem solving in general. For simple problems, the problem space may be small; for complex problems, the number of possible states and thus the size of the network representing the problem space can rapidly become unwieldy. Figure 4.5 shows the problem space for a situation in which each state leads to two possible actions which transform the state into one of a number of further states It can be seen that the number of possible terminal states increases geometrically in proportion to n, where n is the number of operations in a sequence of operations required to reach those states. In both human and computer based systems, heuristics are used to prune the tree of possible states in order to reduce it to a manageable level (Wickelgren, 1984, Rich, 1983)

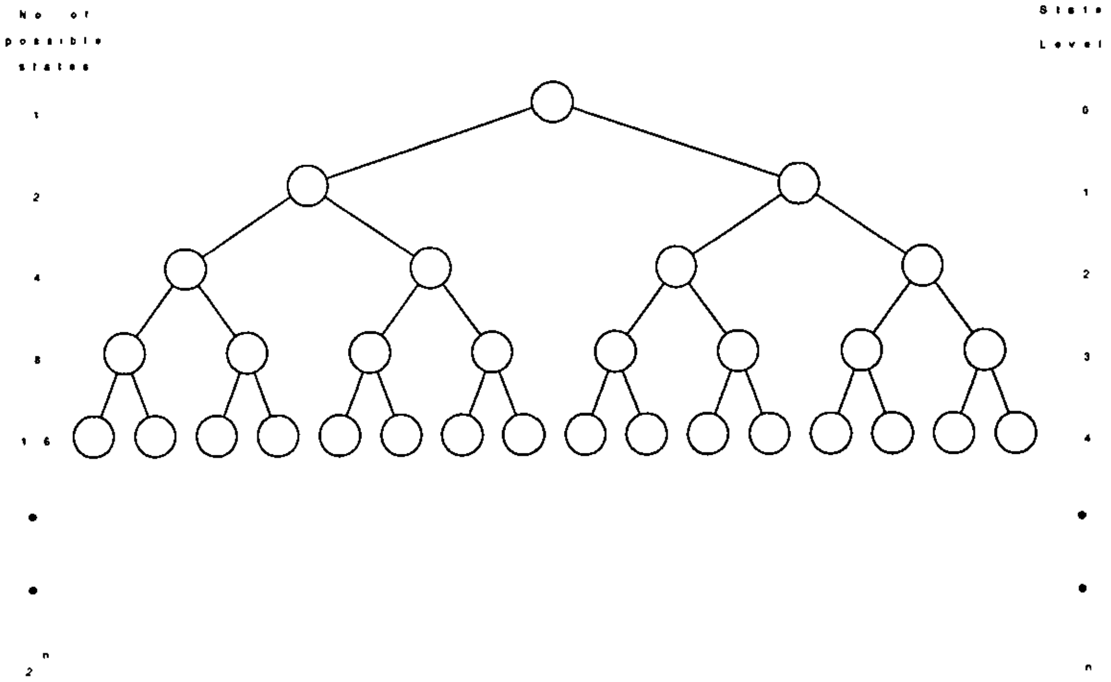


Figure 4.5 *State-action tree for a problem with two possible actions at each state, showing the geometrical increase in the number of states at each level*

While this view of problems and problem solving in terms of search through a problem space is useful for mathematical and logic problems and puzzles in which the number of possible states is finite and computable, it is of less use for the solution of problems which require creative thought. However, both for finite problems with large numbers of possible states, and for creative problems, the way in which information is presented to the problem solver can assist the problem solving process.

Perception

A third area of theory which is of use in considering how people solve problems concerns the issue of perception. Much of the information which humans use in making sense of the world comes to us through our eyes, and the sophistication of the visual element of the perceptual subsystem is highlighted by the difficulty which researchers face in developing computerised approaches to tasks such as recognising different faces, tasks which humans can accomplish from a very young age. Indeed research has found that animal visual sensory organs preprocess information, before it is passed to the brain itself For example, the retinas of animals have been found to contain special detectors which enable the animal to spot movement, edges and slits. Within the visual cortex, cells have been found which respond to edges at different orientations and even to right angles. It is believed that the human visual system contains similar inbuilt mechanisms

Other research, rooted in the traditions of Gestalt Psychology, has considered the way in which humans respond to and organise novel and complex visual stimuli. This has led to the specification of the "laws of Gestalt". These laws are

- the proximity law,
- the similarity law,
- the law of good continuity, and
- the law of closure.

Of particular interest in the context of human problem solving and perception are the proximity law, the similarity law and the law of closure

The proximity law reflects the fact that objects close to one another are perceived as forming a unit or "Gestalt", as shown in Fıgure 4 6(a). The similarıty law reflects the way in which similar objects are perceived as forming a "Gestalt", as shown in Figure 4.6(b). The law of closure descnbes how we fill gaps ın our perception in order to achieve a "Gestalt", as shown in Fıgure 4.6(c) (Wærn, 1989).

The important feature of the way in which humans perceive the world around them is that mechanisms exist which help us to see wholes rather than parts, patterns rather than indıvidual elements.

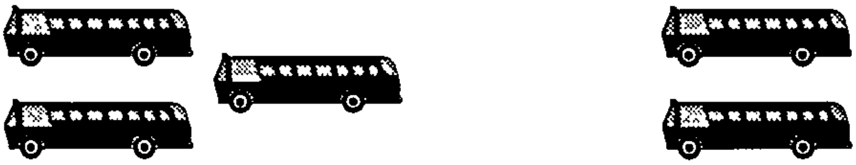


Figure 4.6(a) *Gestalt pıoximıty law*

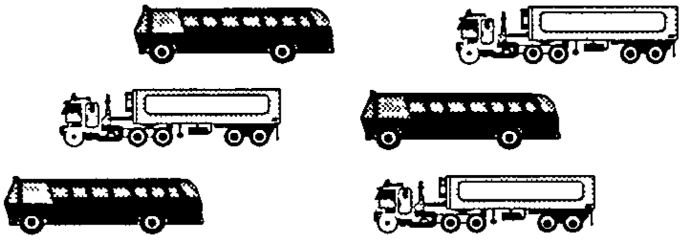


Figure 4 6(b) *Gestalt similarity law*

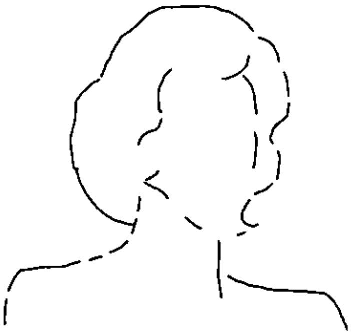


Figure 4 6(c) *Gestalt law of closure*

How Computers Can Help Make Decisions

There have been a number of strands in the development of computer systems designed to assist people in makıng decısıons and solving problems. Three of these are considered here·

- Operations Research,
- Artificial Intelligence,
- Decisıon Support Systems.

Operations Research

The first of these three strands is that of Operations Research (OR), the application of mathematical technıques to solvıng problems. Winston (1987) defines OR as "a scientıfic approach to decision ımaking, which seeks to determine how best to design and operate a system, usually under conditions requiring the allocation of scarce resources". OR developed during the Second World War, as the applicatıon of scıentific techniques to ımilitary problems Today, it is widely used in industry, and makes extensıve use of computerısed approaches, although not exclusıvely, in order to solve problems Winston (1987) outlines a seven-step procedure for the use of OR ın an organısatıon.

- Formulate the problem,
- Observe the system,
- Formulate a mathematical model of the problem,
- Venfy the model and use the model for predictıon,
- Select a suitable alternative,
- Present the results and conclusions of the study to the organısatıon, and
- Implement and evaluate recommendations.

OR uses technıques such as lınear programmıng, queuing theory, and Monte Carlo simulatıon in order to solve organisational problems Most studies of routeing and schedulıng problems are made from an OR perspective, and there are a number of well documented technıques for solving such problems (Bodın et al , 1983). Most such technıques rely on heurıstıc approaches to lımıt the number of possıble combinations which must be considered in an exhaustive search through the problem space.

While OR started as a discipline before the advent of the computer, it has benefited greatly from developments in computer science and the availability of modern computer systems Programs are available which apply many of the techniques of OR.

Artificial Intelligence

The second strand is that of Artificial Intelligence (AI). Rich defines AI as "the study of how to make computers do things at which, at the moment, people are better" (1983). Essentially, this concerns the cognitive activities which go beyond remembering facts and figures and routinely manipulating figures and strings of characters Thus it is easy for a computer to total up the figures in a balance sheet, more difficult for it to interpret the balance sheet and understand the performance of the company It is easy for a computer to check the spelling in a document, at present still impossible for it to understand the content of the document and answer questions which concern the meaning of the text, except in very narrow domains.

Scientists of two types study AI computer scientists interested in the possibility of computers performing tasks which at present can only be performed by people, and cognitive scientists interested in the light which attempting to get computers to perform such tasks throws on the ways in which humans perform them. Thus AI research covers a wide range of areas Examples are:

- Game playing
- Theorem proving
- General problem solving
- Perception
- Natural language understanding
- Expert problem solving

(Rich, 1983). The aim of AI is for the computer to perform the task in a way that is indistinguishable from a human, and this has been a dream since before the development of the first electronic computers, and a dream which is even now only beginning to be achieved at the simplest level.

While AI is very dıfferent from OR ın the techniques which it applıes to problem solving, OR is a catholic discipline, prepared to absorb any technıques whıch contribute to the modelling and solution of problems There is a growing interest in the application of AI techniques wıthin OR ın general (Phelps, 1986) and in relation to varıous types of scheduling problems (Grant, 1986; Knott, 1988).

Decision Support Systems

The third strand of development concerns what are known as Decisıon Support Systems (DSS). Early commercial computer systems dealt with Electronic Data Processing (EDP): the processing of many transactions representıng financial or other events in the busıness, such as sales and purchases, stock receipts and issues, bank wıthdrawals and payments ın, or payroll details. From the early EDP systems, Management Information Systems (MIS) developed as people recognised that the large quantıtıes of data held by EDP systems could be used as the basis for reports to management. Typıcally such reports were predefined and produced at regular intervals: monthly sales reports, weekly stock reports, quarterly wage summarıes. The wider availabılity of computing power and the development of software packages such as sımulation languages and spreadsheets have led since the 1970's to the development of DSS, whıch use the data from EDP systems in a more flexıble way, allowing managers to extract data from a database, feed it into some kind of model of the enterprıse or of part of ıt, and to use the results as the basis for management decısıon makıng. The components of DSS have been described by Sprague and Carlson ın terms of the Dialog-Data-Model Paradigm (1982). The inclusıon of the dialogue element reflects the fact that DSS, unlike MIS should be ınteractıve, allowıng the user to try out various optıons, possıbly bringing data together ın novel ways ın an attempt to devise a solution to a problem, whereas a user of a MIS ıs lımıted to predefıned reports

Sprague and Carlson also provide an approach to the systems analysis process whıch is specıfic to DSS This is the ROMC (Representatıons, Operatıons, Memory Aids, Controls) approach (Sprague and Carlson, 1982) Representations concern the way ın whıch the decısıon maker conceptualises the ınformation needed for the decısıon, for example lists, graphs, plots, tables. Operatıons are those operations which must be performed in order to support the decision, and may be complicated algorithms or simple manipulation of data. Memory aids are used by decision makers to supplement their own memory and may be as simple as jottings made on a scratch pad, while in a computerised DSS they may be workspaces or databases used to hold the transient results of operations. Controls are the mechanisms by which the DSS is operated, such as menus or special keys to press, or which aid the user by providing help or error messages which are easy to understand.

Over the last decade there has been a convergence of DSS with one sub-strand of AI, Expert Systems (ES) (Turban, 1990). ES use the techniques of AI to model the way in which human experts solve problems within a narrow domain. Early examples of ES domains are:

- Medical diagnosis,
- Computer system configuration,
- Mineral prospecting.

Since then, ES have become more widely used and applied to a number of domains However, ES are most applicable in fields where there is a clear distinction between the performance of an expert in the field and that of a novice or lay person, that is to say that there truly is a body of expert knowledge rather than a problem which can be solved by the use of general knowledge and common-sense. Indeed problems requiring the application of common-sense are regarded as unsuitable for the application of expert system techniques, and in the context of vehicle scheduling and routeing, it is interesting to note that Yazdani has suggested that domains where ''the knowledge includes spatial and/or temporal relationships'' are unsuitable for ES solutions (Yazdani, 1984).

While the starting points or motivations of researchers and practitioners in these three fields of the application of computerised techniques to problem solving may be different, they do have in common the intention of applying computers to the solution of problems which face people The catholic and synthetic approach which is reflected by OR and by DSS is one which has been adopted in this research, attempting to apply techniques which are appropriate and which produce the desired results, rather than taking a purist or a reductionist approach

Functions in CT Operations

The first part of this chapter has considered how people make decisions and solve problems and how computers can be used in those processes. This second part considers four functions which are performed in the day to day operations of CT, the ways in which they can be computerised, and the way in which each has been computerised in the software developed during this research. Chapter 3 covered the analysis of the basic data structures required for CT operations. These additional functions are based on those data structures and enhance them to provide the necessary operational features. These facilities are those which go beyond the mere maintenance of the database, in order to provide the functions which are based on the skill of people as booking clerks, dispatchers or drivers.

The additional facilities which are covered here are the following:

- On-line vehicle diary
- Brokerage vehicle search
- Ordering of passenger pick-ups
- Dial-a-ride shift scheduling.

Each of these facilities will be explained in turn, and considered in terms of the three areas outlined at the start of this chapter firstly, how people perform the function; secondly, the range of options available to computerise each function; and thirdly, the approach which has been adopted in the software.

On-line vehicle diary

In CT operations, a vehicle diary is used in a way very similar to the way in which people use diaries generally in order to organise future events It is probably worth considering what are the essential features of a diary as opposed to any other way of organising events

As has been discussed above, there are limitations to the ability of humans to hold many items of information in short-term or working memory and to commit items to and retrieve items from long-term memory. It has also been pointed out that artificial extensions to memory, such as pencil and paper, are used to overcome these shortcomings Given these limitations and the need to organise their home, social and working lives, people use diaries in order to record future events. It may seem obvious to organise this information in temporal sequence, but potentially this information could be stored in some other order: in the order in which future events are notified; in some order related to their location in space (for example, all meetings in Birmingham, in Leicester, in London); or in some order related to the other people involved in the events. People do, however, store information in temporal sequence firstly by date, and then by time within date, and the diary is the established means of doing this. Using a diary, it is possible to answer such questions as:

''What am I doing after lunch?''

''What am I doing tomorrow?''

''What am I doing next week?''

''Can I fit in a meeting in London next Wednesday?''

''Can I fit in a meeting in Birmingham, which will take a whole day, any day next week?''

and,

''What was I doing on 12th July 1990?''

The diary format is less useful for answering questions such as:

''When will I next be in Loughborough?''

''When will I be meeting with David next?''

as these questions require a sequential search through the diary day by day until the required information is found.

Where one person has access to the diary of one or more other people, they can organise events for them. This may be in a situation where a secretary or receptionist books visitors or patients in to see a manager or a doctor, or it may be on a peer basis, whereby staff with access to each other's diaries can book meetings for several people

In the case of the doctor's or dentist's practice, a patient can request to see a specific practitioner or can request an appointment on a particular day or at a particular time for whichever practitioner is available then. This brings us closer to the way in which a diary is used in CT group hire operations. A booking diary is used to organise access to limited resources, which could be rooms in a hotel, dentists or minibuses. Essentially a resource dimension is added to the dimension of time, so that at any time there is a choice of resources which may be allocated. Typically, each resource is discrete and uniquely identifiable. While time is a continuum, it is typically used and allocated in discrete blocks, nights in a hotel or 15 minute slots at a dentist's.

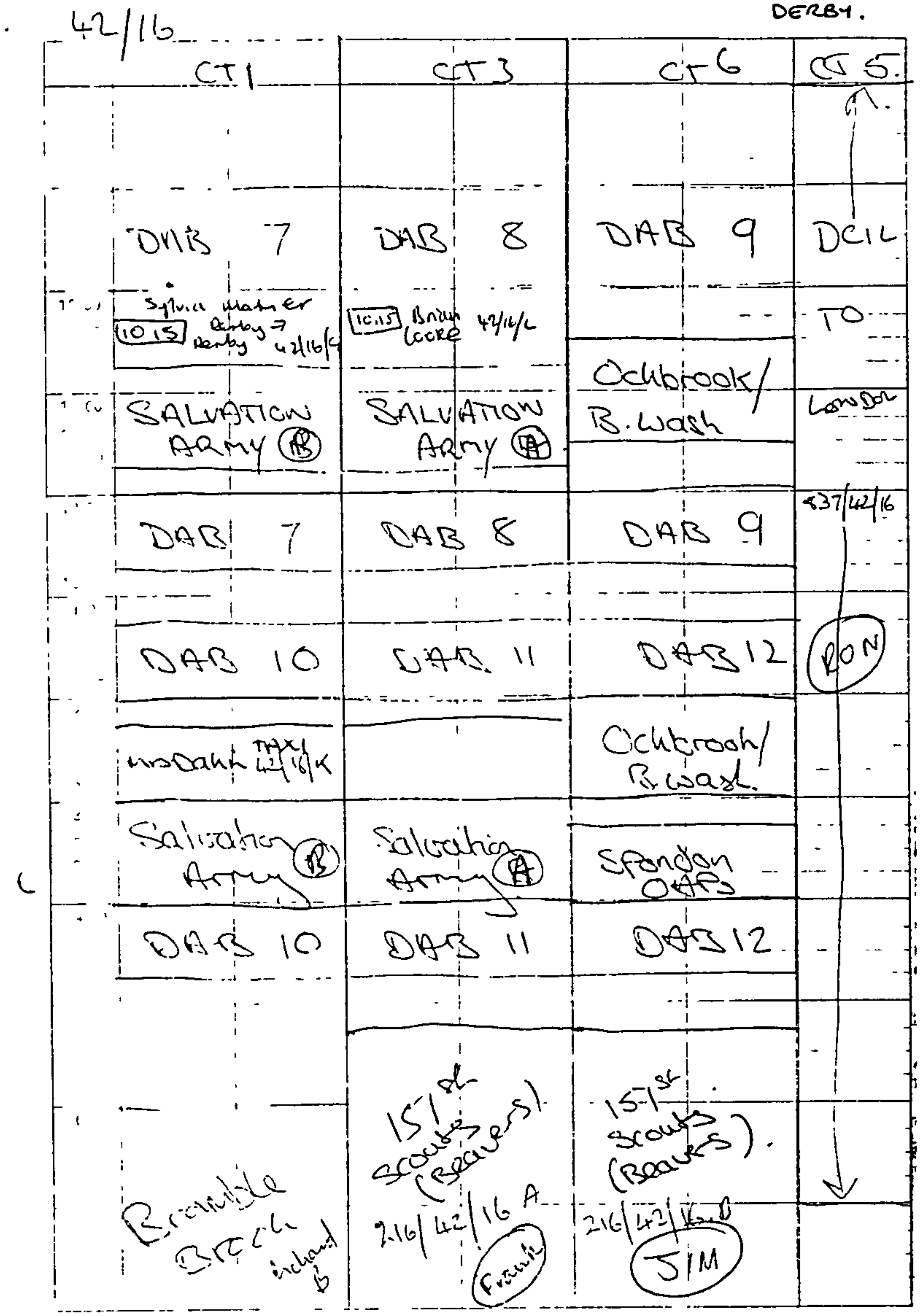
A CT vehicle diary is a booking diary of this type, and examples from Derby CT and Nottingham CT are shown in Figures 4 7 and 4.8. Both of these examples show how vehicles and time appear as two separate dimensions within the diary page. In the Derby case, where the same diary is used for group hire, dial-a-bus and dial-a-ride bookings the time slots are of fifteen minutes duration, while in the Nottingham case, where the diary is for group hire use, and almost all group hire is self-drive, the time slots are morning afternoon and evening. It is important to note that from the point of view of Derby CT which provides more than one service, and where some vehicles can be used for more than one service, a dial-a-bus booking appears in the same way as a group hire booking in the diary. A further example would be Ealing CT, which prior to computerising its group hire operation used a wall mounted board with slots on as a vehicle diary, with a column of slots for each vehicle and over sixty slots in each column to represent two month's worth of bookings. A booking was written onto a card which was placed in the slot for the appropriate vehicle day. Many of Ealing CT's bookings were for the whole day or for more than one day.

A vehicle diary can thus be used to answer similar questions to an individual's diary:

"What is vehicle CT1 doing this afternoon?"

"What is vehicle CT3 doing at 2 00pm on Friday?"

However, the vehicle booking diary is used for specific operational purposes These are, firstly, the ability to check whether a vehicle is in use between certain times on a certain day or days; and, secondly, the ability to see at a glance where

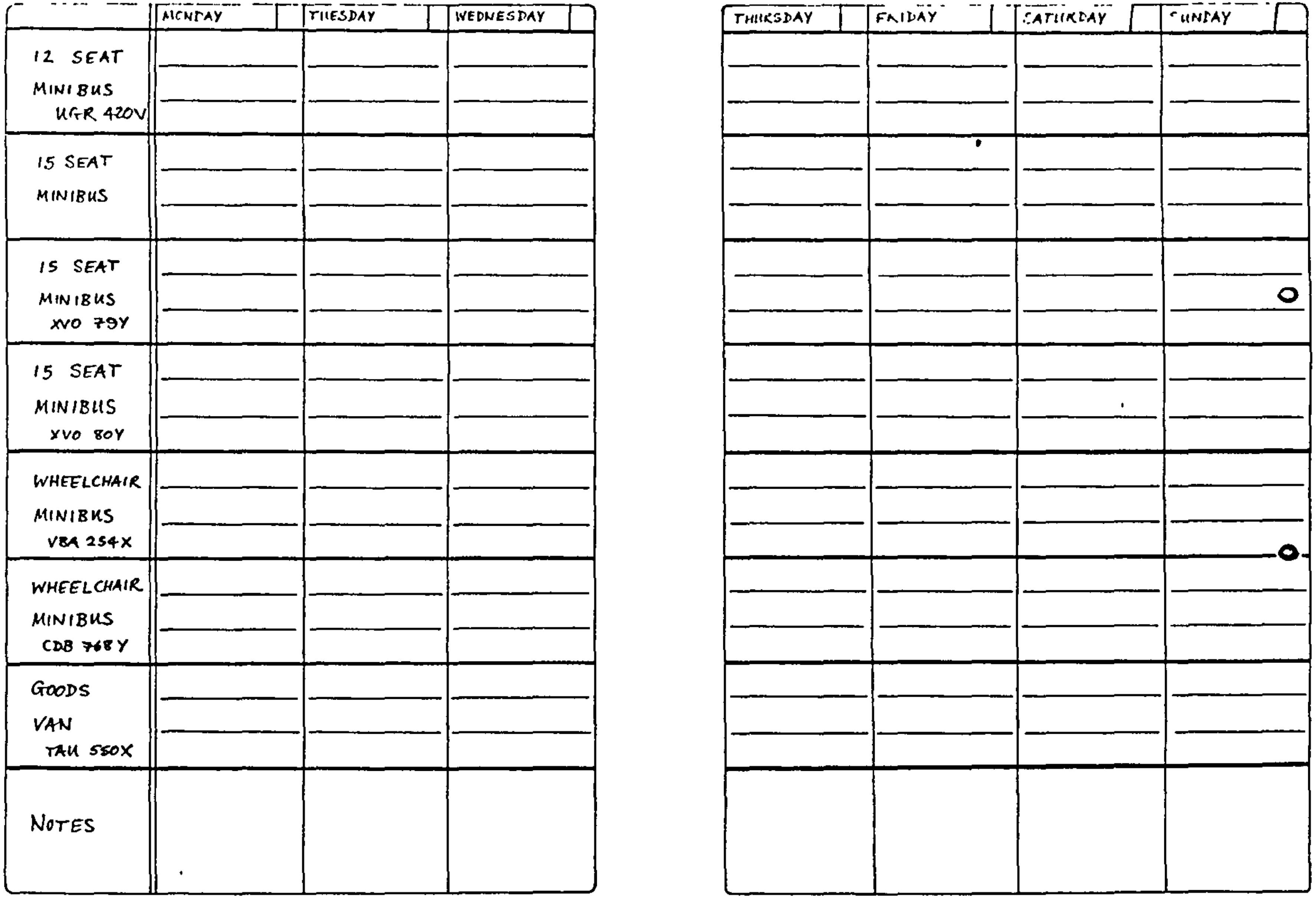


42/16

DERBY.

CT1	CT3	CT6	CT 5.
			↑
DAB 7	DAB 8	DAB 9	DCIL
Sylvia Mather 10.15 Derby → Derby 42/16/G	10.15 Brian Cooke 42/16/L		TO
SALVATION ARMY B	SALVATION ARMY A	Ockbrook/ B. Wash	LONDON
DAB 7	DAB 8	DAB 9	537/42/16
DAB 10	DAB 11	DAB 12	RON
mrs Oakh TAXI 42/16/K		Ockbrook/ B wash.	
Salvation Army B	Salvation Army A	Spondon OAPs	
DAB 10	DAB 11	DAB 12	
Bramble Brech orchard B	151st scouts (Beavers) 216/42/16 A Frank	151st scouts (Beavers). 216/42/16 B JIM	↓

Figure 4.7 *Derby CT booking diary page*



	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY
12 SEAT MINIBUS UGR 420V							
15 SEAT MINIBUS							
15 SEAT MINIBUS XVO 79Y							
15 SEAT MINIBUS XVO 80Y							
WHEELCHAIR MINIBUS VBA 254X							
WHEELCHAIR MINIBUS CDB 768Y							
GOODS VAN TAU 550X							
NOTES							

Figure 4.8 Nottingham CT booking diary page

there is free capacity on a range of vehicles between certain times on certain days.

These two functions might be described as the checking function and the searching function. In the first case, the human has a requirement and has a vehicle in mind; the answer is a yes or a no. In the second case, the human has a requirement and is looking for a vehicle to meet that requirement; the answer is a set of vehicles which could meet that requirement, which may be an empty set.

Manual diary pages typically contain the minimum critical information in order to perform the two functions identified above, and are designed to allow the human user to see the information at a glance. Human users appear to perform this task by a Gestalt process, seeing the pattern of usage and availability rather than establishing the information by a serial search through the diary. It can be seen from the example pages that there is a visual correlation between the duration of a booking and the amount of space taken up in the booking diary, which supports this Gestalt process. This minimum information supports the use of a booking diary by a human user to get a picture of usage at a glance, but it does not provide the additional information, which a user may require about the booking, such as the number of passengers. Typically, additional information about bookings is held on booking sheets, held in a ring-binder or some similar filing system, and one of the inherent weaknesses of manual systems is the need to cross-reference different sources of information.

In CT operations, the use of a vehicle booking diary supports the taking of bookings from users. It provides the facilities to allow the user to undertake the checking function and the searching function. However, there are constraints on booking vehicles, such as the number of passengers or the type of facilities required for passengers with disabilities. These will be discussed further in the section on brokerage vehicle search. The searching function described here may indicate that there is no suitable vehicle available at the time required, although other vehicles may be available. One of the aims of most CTs is to maximise vehicle usage, and people taking bookings will usually consider moving an existing booking from a suitable vehicle in order to accommodate a new booking, if the existing booking can be allocated to a vehicle which is suitable for it but unsuitable for the new booking.

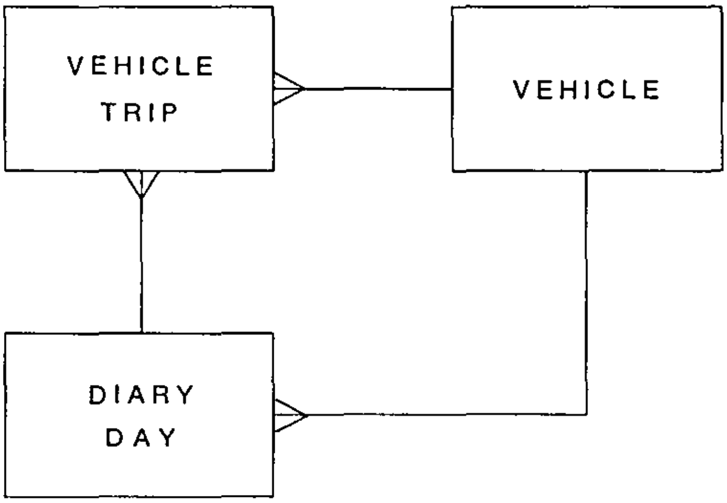
This process can be considered in terms of a search through a problem space The initial state consists of a new booking which has not been allocated to a vehicle, a set of vehicles and a set of existing bookings allocated to those vehicles, which relationship is recorded in the diary. The goal state is for all bookings, including the new one to be allocated to vehicles. Valid operations are to allocate a booking to a vehicle and to unallocate a booking from a vehicle or alternatively to swap a booking from one vehicle to another. Attempting to swap a booking from one vehicle to another may create a further problem involving the need to swap a further booking, so the process can be recursive. This process is similar to approaches to problems like the 8-puzzle, the Tower of Hanoi or transporting a fox, a rabbit and a lettuce across a river in a boat, which are so popular with the artificial intelligence community. While the search space for such problems can grow exponentially, the search space for this problem in the context of CT operations is naturally bounded by the requirement that any swap results in a satisfactory allocation of a booking to a vehicle, that is in terms of capacity and facilities

CT operators can be observed going through this process in most group hire operations, often thinking aloud as they do so:

"The Stroke Club need a vehicle all day, and it has to have a tail-lift. We can't give them CT3 because it's in use for Dial-a-Bus in the afternoon, and for Salvation Army in the morning We can't give them CT2 because it's being used by the Sikh Temple in the morning. CT6 is free, but it hasn't got a tail-lift. We could move Salvation Army onto CT6, because they can manage without a tail-lift Then we could move the Sikh temple onto CT3, so the Stroke Club could have CT2."

This kind of situation is easier for people to manage in the kind of operation which has few vehicles, takes bookings by the morning, afternoon or evening or whole day, or does not run multiple services on the same vehicles. The information processing model of the human problem solver implies that the limit to the number of pieces of information which can be held in working memory limits the number of possible alternatives which a human CT operator can consider when attempting to allocate a new booking to a vehicle. Some of the information which the operator uses is held in long term memory, for example whether a particular vehicle has a tail-lift, and will require little effort to recall, while other information will be held in working memory, for example the number of passengers to be carried for the current booking, and will require mental effort to retain in working memory. The process of allocating bookings must be supported not just by the diary, but also by the availability of booking records which contain additional information. The following section outlines how this process is supported using the diary facilities of the software.

The data structure of the vehicle diary has been described briefly in Chapter 3. A record may be held in the diary for every vehicle for every day. In fact an entry is only made when a vehicle trip is added to the diary for that vehicle-day. Each record in the diary contains details of trips for that vehicle on that day in chronological sequence. The diary entries hold some of the same information as is held in the trip records. This redundancy is introduced deliberately in order to allow rapid checking of the availability of a vehicle between particular times on a particular day or days. The minimum logical data structure is shown in Figure 4.9 as an E-R diagram with attributes for the DIARY entity. The physical data contents are shown in Table 4.1 with sample data in PICK internal and external format.



DIARY DAY
(date,
vehicle no,
vehicle trip no)

Figure 4.9 *Logical data structure of diary*

The physical data structure reflects the way that data is held in a PICK database, and is implementation dependent. It would not be held in the same way, for

Vehicle No	0001			
Date	8619 06/08/1991			
Booking No	001234	005796	7697/1/E	005678
Trip Number	0001567	0002765	0002145	0001256
Start Time	28800 08 00	36900 10 15	41400 11 30	54000 15 00
Finish Time	36000 10 00	40500 11 15	53100 14 45	72000 20 00
Service Code	O	G	D	G
User Id	0000	0156	DAR	0217

Table 4.1 *Physical data contents of diary*

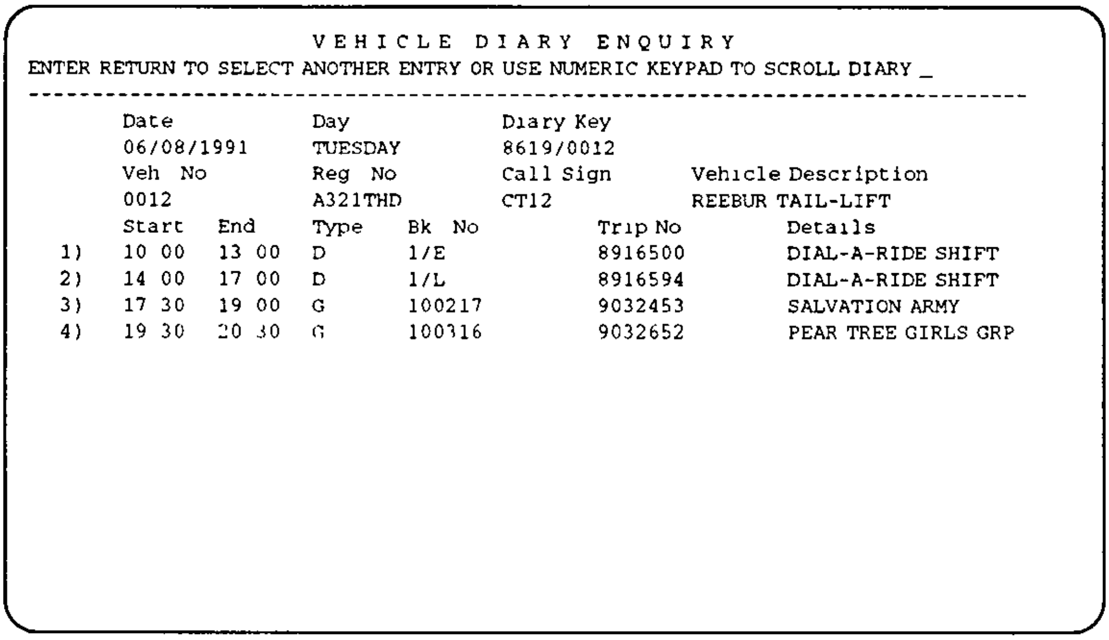
example, in a true relational database.

The checking and searching functions are provided to users as part of the software. The first is provided as a subroutine that can be called from a booking entry or amendment screen At the point that the user enters a vehicle number into the screen, the subroutine is called with the parameters from the current trip which are required in order to check whether it can be fulfilled on that vehicle. The subroutine reads the entry or entries from the diary for the day or days concerned and checks using a relatively simple algorithm whether the new trip can be allocated to that vehicle. It returns an error number which indicates whether the trip can be allocated and if not, why not, for example because the new trip overlaps an existing trip.

If it is not possible to allocate the new trip to that vehicle, the software offers the user the option to view the diary for that vehicle day, allowing an entry point into the second function The software also checks that the vehicle has the seating and wheelchair carrying capacity to carry the passengers for that trip, and that it has any required facilities, such as a tail-lift or a ramp.

The search function can be performed by the user viewing the diary for a particular vehicle day The screen layout is shown in Figure 4.10.

This screen can be viewed as a window on the overall diary, and can be moved by the user with the keys on the numeric keypad on the keyboard. The conceptual view of this window is shown in Figure 4.11. The user can view the other vehicles for that day, or can move the window to other days, either forwards or backwards in time (a day, a week or a month at a time). This facility is a part of a more complete diary search function.



```
                         V E H I C L E   D I A R Y   E N Q U I R Y
ENTER RETURN TO SELECT ANOTHER ENTRY OR USE NUMERIC KEYPAD TO SCROLL DIARY _
-------------------------------------------------------------------------------
        Date              Day             Diary Key
        06/08/1991        TUESDAY         8619/0012
        Veh  No           Reg  No         Call Sign       Vehicle Description
        0012              A321THD         CT12            REEBUR TAIL-LIFT
        Start    End      Type     Bk  No          Trip No         Details
   1)   10 00    13 00    D        1/E             8916500         DIAL-A-RIDE SHIFT
   2)   14 00    17 00    D        1/L             8916594         DIAL-A-RIDE SHIFT
   3)   17 30    19 00    G        100217          9032453         SALVATION ARMY
   4)   19 30    20 30    G        100316          9032652         PEAR TREE GIRLS GRP
```

Figure 4 10 *Sci een layout of diary for a vehicle day*

The more complete diary search function starts in a screen which shows the availability of five vehicles over three days for each service This screen is shown in Figure 4.12. (An enhancement was subsequently made during 1990 which provides an additional diary screen which shows 10 vehicles over a fourteen day period in less detail. This is shown in Figure 4 13.)

Within this screen the user can highlight a particular vehicle day by moving a highlighting cursor with the numenc keypad. It is then possible to zoom into the diary screen descnbed above (and shown in Figure 4.10) for that vehicle day, in order to view more detail of the bookings for that day.

The user can select and highlight a particular trip in the diary screen and then perform a number of functions in relation to that trip, most of which are outlined below. The most significant from the point of view of the search function is the ability to view the full details of any vehicle tnp in an enquiry screen for the particular type of tnp concemed.

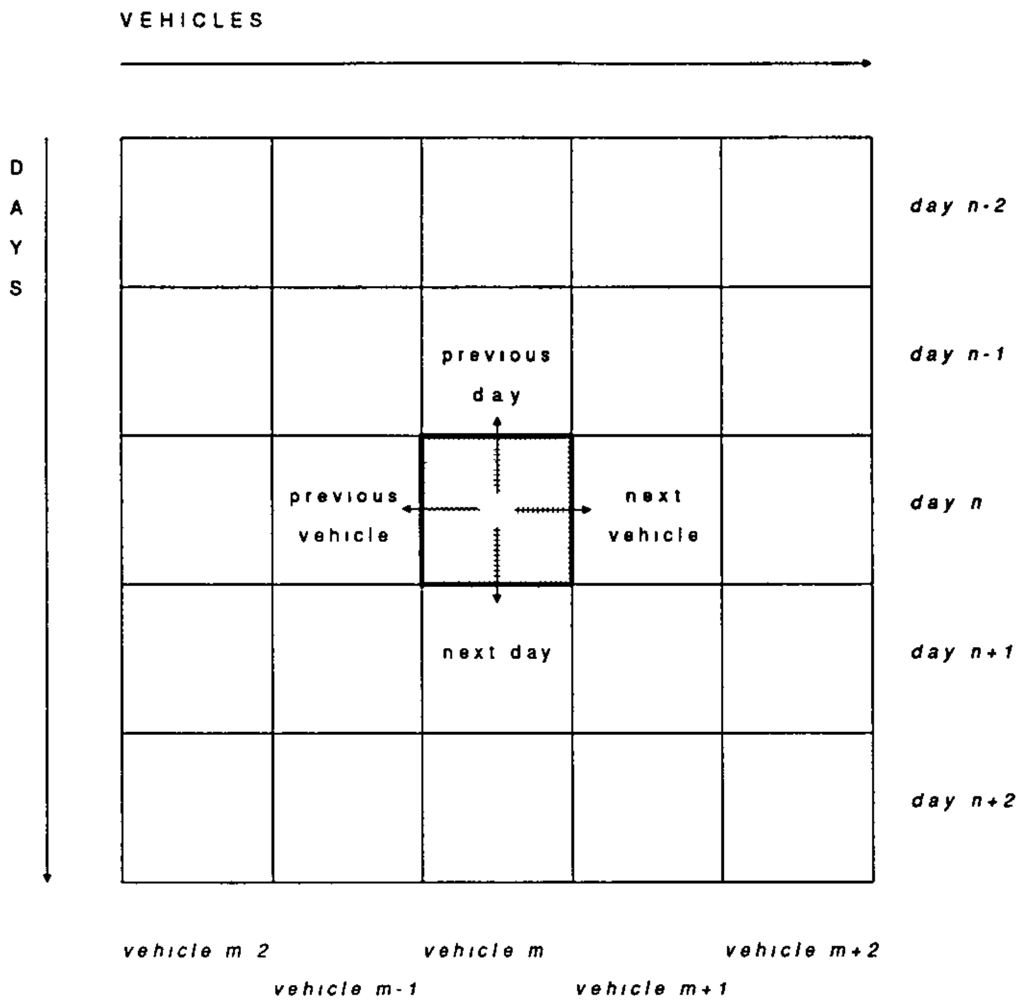
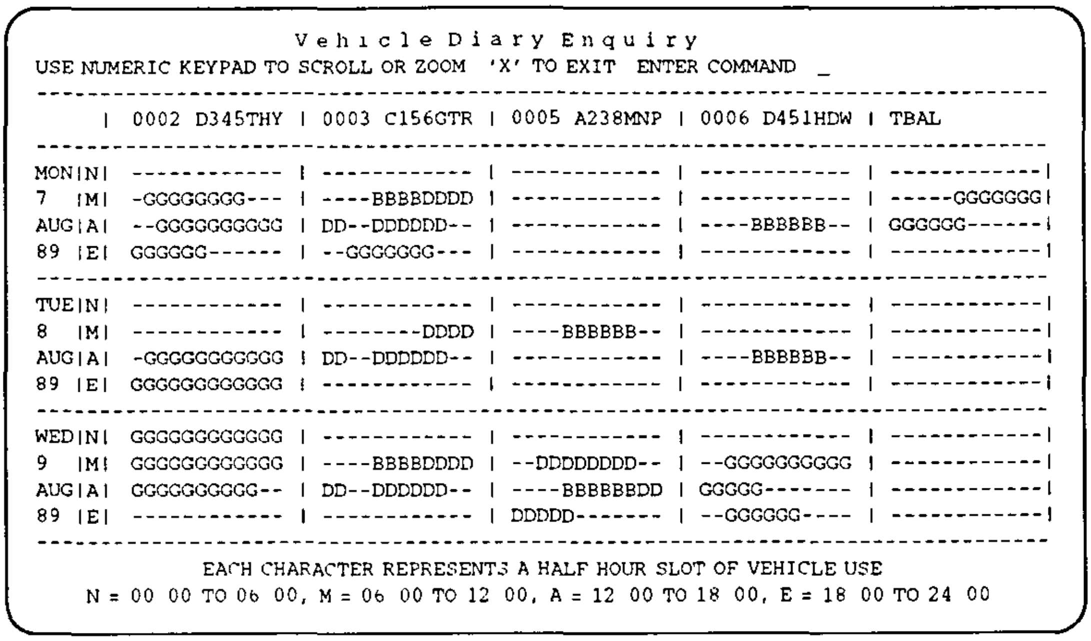


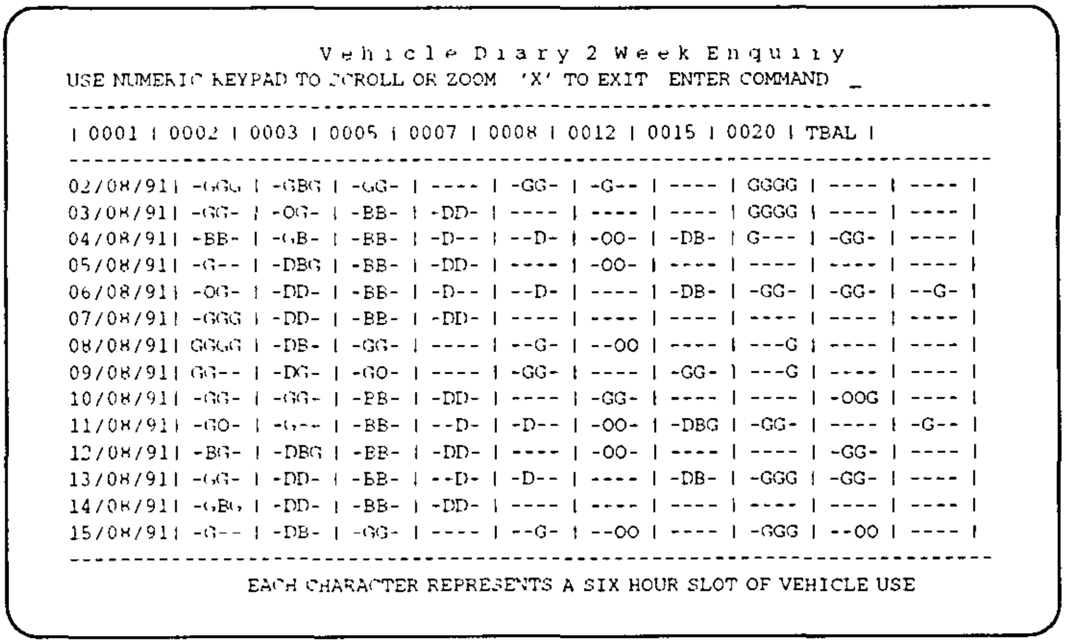
Figure 4.11 *Conceptual view of diary screen as a window on a range of vehicles and days*

A further function that is performed by human users of diary systems, in attempting to allocate a booking, is the "What If?" function. Given a pattern of vehicle availability, which the user can see, the user may consider moving an existing trip to another vehicle in order to make room for the new trip, as has been described above. This is dependent on the user having available to him or her information about the requirements for each trip, for example the need for special facilities such as a tail-lift or for a vehicle with a certain capacity. In a manual system, the diary typically holds the minimum information, and detailed information is held in files of booking forms which the user may have to fetch to his or her desk. In the computerised system, this is available directly, by moving to the appropriate vehicle day, selecting and highlighting the relevant trip and zooming in on it.



```
                    Vehicle Diary Enquiry
USE NUMERIC KEYPAD TO SCROLL OR ZOOM  'X' TO EXIT  ENTER COMMAND  _
------------------------------------------------------------------------------
     | 0002 D345THY | 0003 C156GTR | 0005 A238MNP | 0006 D451HDW | TBAL
------------------------------------------------------------------------------
MON|N|  ------------ | ------------ | ------------ | ------------ | ------------|
7  |M|  -GGGGGGG--- | ----BBBBDDDD | ------------ | ------------ | -----GGGGGGG|
AUG|A|  --GGGGGGGGGG | DD--DDDDDD-- | ------------ | ----BBBBBB-- | GGGGGG------|
89 |E|  GGGGGG------ | --GGGGGGG--- | ------------ | ------------ | ------------|
------------------------------------------------------------------------------
TUE|N|  ------------ | ------------ | ------------ | ------------ | ------------|
8  |M|  ------------ | --------DDDD | ----BBBBBB-- | ------------ | ------------|
AUG|A|  -GGGGGGGGGGG | DD--DDDDDD-- | ------------ | ----BBBBBB-- | ------------|
89 |E|  GGGGGGGGGGGG | ------------ | ------------ | ------------ | ------------|
------------------------------------------------------------------------------
WED|N|  GGGGGGGGGGGG | ------------ | ------------ | ------------ | ------------|
9  |M|  GGGGGGGGGGGG | ----BBBBDDDD | --DDDDDDDD-- | --GGGGGGGGGG | ------------|
AUG|A|  GGGGGGGGGG-- | DD--DDDDDD-- | ----BBBBBBDD | GGGGG------- | ------------|
89 |E|  ------------ | ------------ | DDDDD------- | --GGGGGG---- | ------------|
------------------------------------------------------------------------------
              EACH CHARACTER REPRESENTS A HALF HOUR SLOT OF VEHICLE USE
    N = 00 00 TO 06 00, M = 06 00 TO 12 00, A = 12 00 TO 18 00, E = 18 00 TO 24 00
```

Figure 4 12 *Diary for 5 vehicles over 3 days*



```
                    Vehicle Diary 2 Week Enquiry
USE NUMERIC KEYPAD TO SCROLL OR ZOOM  'X' TO EXIT  ENTER COMMAND  _
------------------------------------------------------------------------
| 0001 | 0002 | 0003 | 0005 | 0007 | 0008 | 0012 | 0015 | 0020 | TBAL |
------------------------------------------------------------------------
02/08/91| -GGG | -GBG | -GG- | ---- | -GG- | -G-- | ---- | GGGG | ---- | ---- |
03/08/91| -GG- | -OG- | -BB- | -DD- | ---- | ---- | ---- | GGGG | ---- | ---- |
04/08/91| -BB- | -GB- | -BB- | -D-- | --D- | -OO- | -DB- | G--- | -GG- | ---- |
05/08/91| -G-- | -DBG | -BB- | -DD- | ---- | -OO- | ---- | ---- | ---- | ---- |
06/08/91| -OG- | -DD- | -BB- | -D-- | --D- | ---- | -DB- | -GG- | -GG- | --G- |
07/08/91| -GGG | -DD- | -BB- | -DD- | ---- | ---- | ---- | ---- | ---- | ---- |
08/08/91| GGGG | -DB- | -GG- | ---- | --G- | --OO | ---- | ---G | ---- | ---- |
09/08/91| GG-- | -DG- | -GO- | ---- | -GG- | ---- | -GG- | ---G | ---- | ---- |
10/08/91| -GG- | -GG- | -PB- | -DD- | ---- | -GG- | ---- | ---- | -OOG | ---- |
11/08/91| -GO- | -G-- | -BB- | --D- | -D-- | -OO- | -DBG | -GG- | ---- | -G-- |
12/08/91| -BG- | -DBG | -BB- | -DD- | ---- | -OO- | ---- | ---- | -GG- | ---- |
13/08/91| -GG- | -DD- | -BB- | --D- | -D-- | ---- | -DB- | -GGG | -GG- | ---- |
14/08/91| -GBG | -DD- | -BB- | -DD- | ---- | ---- | ---- | ---- | ---- | ---- |
15/08/91| -G-- | -DB- | -GG- | ---- | --G- | --OO | ---- | -GGG | --OO | ---- |
------------------------------------------------------------------------
              EACH CHARACTER REPRESENTS A SIX HOUR SLOT OF VEHICLE USE
```

Figure 4.13 *Diary for 10 vehicles over 14 days*

In order to support the "What If?" function two further facilities are available.

By highlighting a particular vehicle trip and entering 'A' for Amend, the user can be taken directly into an amendment screen, in which an existing booking can be amended by allocating it to a different vehicle. When it is written to file, the diary is also updated and redisplayed.

By highlighting a particular vehicle trip and entering 'M' for Move, the user can attempt to move a trip to another vehicle day. The trip details are displayed at the bottom of the screen, and the user can select a different vehicle day and then enter 'I' for Insert. The software then attempts to allocate that trip to the new vehicle on the new day (either of which may be the same as the original vehicle day). The same checks on passenger capacity and facilities are made as when the booking is first entered, relieving the human user of worrying about whether the trip can feasibly be allocated to that vehicle. If it is not possible to allocate the trip to the chosen vehicle, a reason is displayed as an error message, and the user can continue, trying other vehicles. The user also has the option of allocating a booking to a dummy vehicle, "TBAL", To Be Allocated Later. This allows existing allocated bookings to be unallocated and held in limbo while other bookings are moved around until a satisfactory solution is found The computer relieves the user of the need to retain in working memory a list of bookings which have been unallocated and which must subsequently be reallocated. It also relieves the user of the need to hold in working memory much of the information about the current booking which is necessary in order to determine whether it can be allocated to a particular vehicle.

Finally, further facilities have been provided which make it possible to operate the entire booking system from the diary screen without having to use the menus around which the system is built. These are:

An insert function to allow a booking or shift entry for any service to be made directly from within the diary

A regular insert function to allow a regular booking or repeat shift entry for any service to be made directly from within the diary.

A log sheet entry or amendment function to allow vehicle log sheets to be entered directly from within the diary

These functions allow the user to operate the system from the diary screen, moving from vehicle to vehicle and from day to day in order to enter, amend and view vehicle bookings and to enter log sheet details into trip records.

Brokerage vehicle search

The difference between the booking process for brokerage operations and standard group hire operations has already been highlighted in Chapter 3, and data flow diagrams included to show the difference in approach. The standard group hire booking facility within the software was designed in order to support both types of operation by the introduction of a dummy vehicle TBAL (To Be Allocated Later), mentioned above in the context of the diary

In a small brokerage operation, the person taking the bookings is likely to know all the vehicles and be familiar with their availability, and it is possible to allocate a vehicle to a booking from memory, or from a simple list of available vehicles. Using a computerised system, such as the package developed here, it is possible to use the booking screen and diary facilities in order to identify a suitable brokerage vehicle for a trip In larger operations, where there may be over a hundred vehicles belonging to many organisations in brokerage, and where these vehicles are located over a wide area, as in the case of Birmingham Shared Transport Services (Warrington, 1986), then it is more difficult to select a vehicle for a particular trip.

It is important to consider that CT vehicle brokers have objectives which go beyond the dictionary definition of the term as a 'middleman' or 'intermediary'. Such objectives are usually concerned with increasing or rationalising the use of under-used vehicles as resources for voluntary organisations (Warrington, 1986; Bainbridge (ed), 1987). Moreover, the intention is to provide a good match between vehicle users and vehicles in terms of location and suitability of the vehicles.

This is clearly an area in which computerised software can assist the bookings clerk in identifying a suitable vehicle. Study of the brokerage process at Birmingham STS, based on interviews with staff, indicated that two types of criteria are used when assigning a vehicle to a trip booking These are

- Availability Criteria
- Sequencing Criteria

Availability criteria are used to decide whether a vehicle is either available or suitable for a particular trip. They can be answered by a binary yes/no response These criteria answer the following questions.

Is the vehicle already in use? For example, it has already been booked by another group, or is being serviced.

Is the vehicle normally used by the owning group at the time of the proposed trip? For example, the Red Cross always use their minibuses from 10.00 am to 4.00 pm on Tuesdays and Thursdays for their luncheon club.

Is the vehicle adapted to meet the requirements of the booking. For example, does it have a tail-lift?

Does the vehicle have the seated passenger and wheelchair capacity required for the trip?

Are there restrictions on the types of passenger who may be carried in the vehicle, which would exclude the passengers on the trip which is being allocated? For example, minibuses owned by organisations such as Age Concern may only be available in brokerage to carry elderly people as passengers.

Sequencing criteria are used to determine which of the available and suitable vehicles is the most appropriate for the trip booking in question These include the following issues

How far is the base of the vehicle from the starting point of the trip? Groups are usually charged on a mileage basis, and ideally a vehicle should be allocated which involves the minimum mileage overhead

How "difficult to book" is the vehicle? This is designed to ensure that vehicles with many constraints on their availability are considered right from the start when making bookings, rather than left until last, when there is less likelihood of matching them with a trip.

In a manual system, these criteria are assessed by the person taking the booking or allocating a vehicle to the booking. They are assessed approximately, either using a map to assess the distance, or using a heuristic approach to judge how difficult a vehicle is to book Both these two sequencing criteria are discussed in more detail below in the context of a computerised approach.

As with the vehicle diary described above, there are no established computerised techniques for performing the process described here. The algorithm is relatively straightforward and consists of a search through the database for vehicles which meet the availability criteria outlined above in the sequence which the sequencing criteria require for that particular trip, that is, dependent on the start point of the trip

All the functions described as availability criteria are already required in order to validate a group hire booking When a vehicle is selected for a trip booking, a check is made to answer each of the questions listed above. A negative answer may result in forced rejection of the booking, an opportunity to change the requirements, an opportunity to use the diary facility to move another booking to a different vehicle in order to free up the required vehicle, or the opportunity to override the negative response. From a design point of view, it is possible to use the same modular program code in order to perform the checking function in the booking screen and in the brokerage search

The sequencing criteria are not provided elsewhere in the system and must therefore be implemented specifically for the brokerage search facility.

The two sequencing criteria are provided in two different ways. Firstly, the criterion of how difficult a vehicle is to book is catered for by an attribute in the vehicle record called the Sequence Number. This requirement was identified very early in the analysis by Birmingham STS, although its purpose was not clear to many other collaborators and users. The Sequence Number is a number in the range 0-99 which acts as an index to the difficulty of booking a particular vehicle. A low value represents a vehicle which is easy to book; a high value represents a vehicle which is difficult to book A vehicle is difficult to book if it has restrictions on the types of passengers who can be carried in it, if it is used extensively by the group which owns it, and is therefore only available sometimes for brokerage, or if it has a limited or fixed seating capacity A value is assigned to this attribute for each vehicle by the operator.

The purpose of using the Sequence Number is to ensure that vehicles which are difficult to book are considered before vehicles which have fewer restrictions If the vehicles are sorted by descending value of Sequence Number and presented to the user in that order, then he or she will consider the difficult to book vehicles before the others This is designed to reduce the likelihood that later bookings have to be rejected because only vehicles with many restrictions are available. In effect, it forces the user of the system to match trip booking requirements with vehicle facilities rather than choosing the easy approach of using vehicles with few restrictions.

Secondly, the criterion of how far the vehicle base or garage is from the start point of a trip is designed to be calculated from the Map Reference attribute held in the vehicle record. The system is designed to cater for map references in a number of formats. These include the following

- The Geographers' A-Z street plans use alphabetic characters for the x-axis and numbers for the y-axis, with the origin positioned at the top left hand corner. The letters I and O are usually excluded, and beyond 24 grid squares on the x-axis, carry on with Aa Ba Ca etc. a kind of base 24

- The Geographers' A-Z street atlas (the familiar London A-Z) uses the same approach with letters for the x-axis and numbers for the y-axis with a top left origin, but this starts afresh on each page Pages are numbered from the top left of the area covered by the map, in rows, to the bottom right

- Geographia uses the same approach as the Geographers' Map Company in its street plans, but includes the letter O

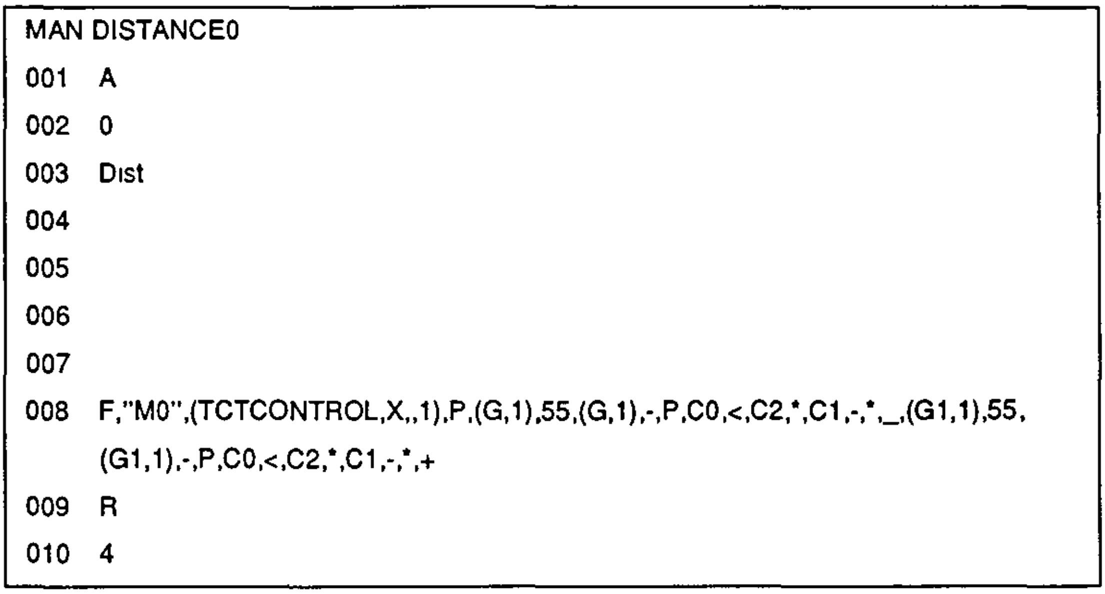
There are many other possibilities, combining letters and numbers or just numbers, on single sheets or on separate pages, with different size grids, and even in the case of Barnsley Metropolitan Borough's Official Street and Area Plan, grid rectangles rather than squares, with different size rectangles on different pages of the map.

In order to calculate an approximate distance between map reference points, a relatively simple approach was taken. It was decided to use Pythagoras' theorem to calculate the distance between grid centres. This was done firstly because the distance only needs to be approximate, and secondly because a straight line has been shown to be as good an approximation as any other for this purpose (Cooper, 1983). In order to use the grid coordinates to perform this calculation, it is necessary to convert the map references to numeric x and y values. This is done when the map reference is entered into a vehicle record, and both alphanumeric map reference and numeric x-y coordinates are stored in the vehicle record. This eliminates the overhead of performing the conversion when the vehicle records are sorted. In the case of the London A-to-Z, or other book based maps, it creates a grid which covers the whole of London of 80 horizontal units by 70 vertical units, and each grid unit being 0.5km. The system is thus capable of being used across the whole of London, or in the proposed regional dial-a-rides for the capital.

The distance for any particular case, however, depends on the location of the start point of the trip. This is entered using the same map reference system at the time that the operator starts the search for an appropriate vehicle, and is converted into numeric x-y coordinates. The distance can then be calculated for each particular vehicle and used as a second sort key.

A design decision was made to use the ACCESS enquiry language to perform the sorting of the vehicle records. The sort routine is written in assembler and runs faster than a sort routine which could be written in BASIC. The problem with this lies in calculating the straight line distance between each vehicle base and the trip start point. This could be done in the BASIC programming language, but it would involve a pass through all the vehicle records calculating the distance and writing the results into the vehicle records before performing the sort. This would cause problems if more than one user was performing a search at the same time. The ACCESS language is dictionary driven. Each field in a record can be given a name, which is held in a dictionary and allows reference to that field for selection or sorting purposes. The dictionary can include names for fields which are not physically held on the file but which are either on other files or are calculated values. Using this facility, it is possible to write what is called an F-Correlative, which uses a stack-based Reverse Polish notation, to perform the calculation. The only restriction is that the F-Correlative notation does not include the facility to take the square root of a number. However, sorting by the square of the hypotenuse has the same effect as sorting by the hypotenuse, so this is not a problem. The dictionary entry for this is shown in Figure 4.14.

The overall brokerage search process is in two parts. First, the vehicle records are

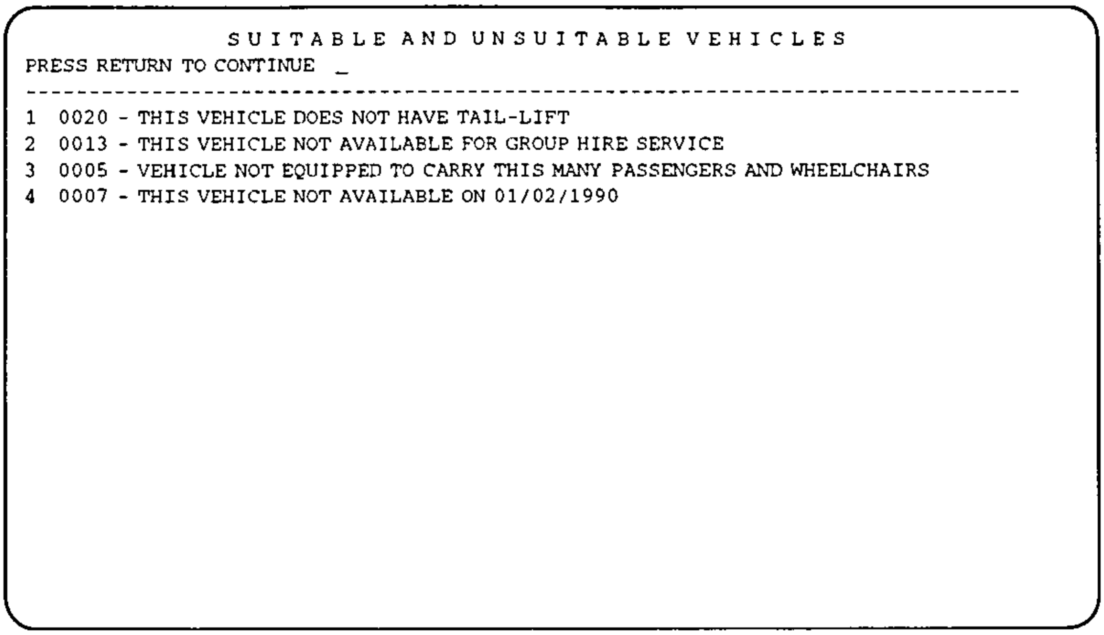


```
MAN DISTANCE0
001  A
002  0
003  Dıst
004
005
006
007
008  F,"M0",(TCTCONTROL,X,,1),P,(G,1),55,(G,1),-,P,C0,<,C2,*,C1,-,*,_,(G1,1),55,
     (G1,1),-,P,C0,<,C2,*,C1,-,*,+
009  R
010  4
```

Figure 4.14 *PICK data dıctıonary entry for sorting vehicles by dıstance from start map reference of bı okerage tı ıp*

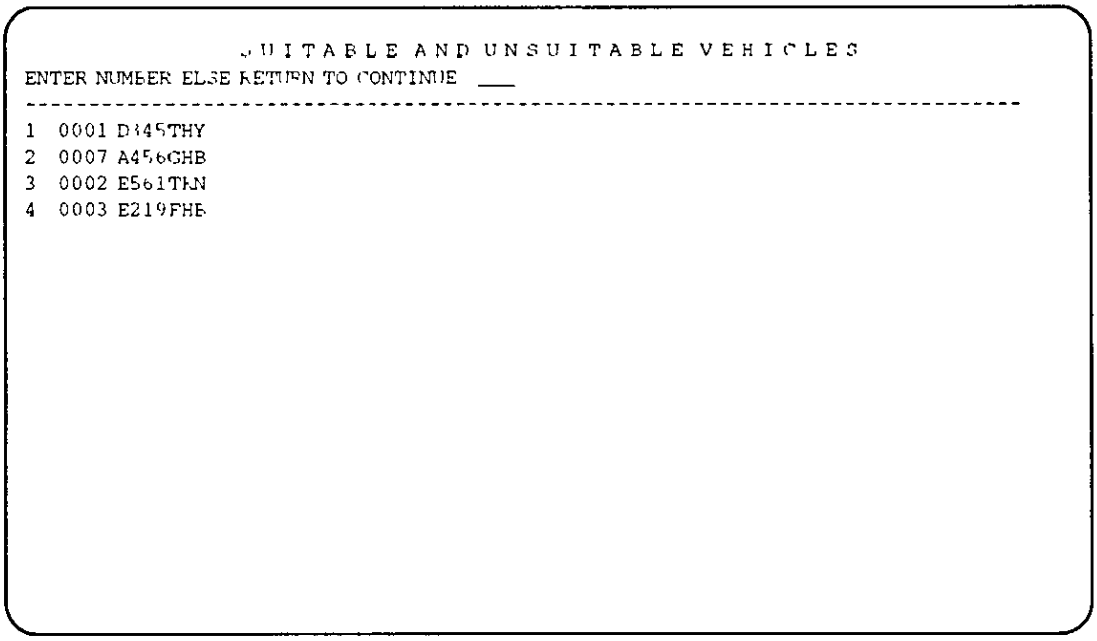
sorted by Sequence Number and by Dıstance. Second, each record is read from the file ın turn, and the availability criteria applied to that vehicle. As soon as it becomes apparent that a vehıcle ıs not either available or suıtable, it is added to a list of unsuıtable vehıcles wıth a summary reason. If ıt passes all the cnteria, it is added to a list of suıtable vehicles. Thıs process is sımılar to a Prolog process ın whıch an attempt ıs made to instantıate a particular vehıcle to a vanable. When all the vehicles have been checked, the user is gıven the option to view the list of unsuıtable vehıcles, or the lıst of suıtable vehıcles, whıch are displayed on screen with theır address and contact telephone number. A vehıcle can be selected from thıs screen and passed back to the bookıng entry screen, by enterıng the number which is dısplayed alongsıde ıt on screen Fıgures 4 15 and 4.16 show the screen dısplays for the vehicle lists

Thıs approach to the brokerage vehıcle search can be vıewed in terms of Sprague's and Carlson's ROMC approach outlined above The decision-maker conceptualises the problem in terms of the vehicles which are available and those which are not. The operatıons ınvolved concern the sorting of vehicles and the checking of the avaılabılıty of each one The database acts as a memory aid, holdıng the required data about each vehicle The system is controlled by entering the command "FIND" instead of a vehıcle number ın the bookıng screen, and the results of the search can be placed back ın the booking screen at that point.



```
                SUITABLE AND UNSUITABLE VEHICLES
PRESS RETURN TO CONTINUE _
----------------------------------------------------------------------------
1  0020 - THIS VEHICLE DOES NOT HAVE TAIL-LIFT
2  0013 - THIS VEHICLE NOT AVAILABLE FOR GROUP HIRE SERVICE
3  0005 - VEHICLE NOT EQUIPPED TO CARRY THIS MANY PASSENGERS AND WHEELCHAIRS
4  0007 - THIS VEHICLE NOT AVAILABLE ON 01/02/1990
```

Figure 4 15 *Unsuitable vehicles for a vehicle brokerage trip*



```
                SUITABLE AND UNSUITABLE VEHICLES
ENTER NUMBER ELSE RETURN TO CONTINUE ___
----------------------------------------------------------------------------
1  0001 D345THY
2  0007 A456GHB
3  0002 E561TKN
4  0003 E219FHB
```

Figure 4 16 *Suitable vehicles for a vehicle brokerage trip*

Ordering of passenger pick-ups

Within the range of CT operations, there are situations in various services which require the ability to organise the picking up of passengers into the most efficient sequence. Examples of this are:

- group hire - picking up passengers from their homes for a trip or to deliver to a common destination;
- dial-a-bus - picking up passengers within a zone to carry to a central destination

In these situations, there is no requirement to order the passengers in time, as there is in dial-a-ride. The only requirement is to sequence the pick-ups in order to minimise the distance travelled by the vehicle This kind of problem is a version of what is known in operations research as the travelling salesperson problem (TSP).

The travelling salesperson problem is easy to state. A salesperson, starting in one city wishes to visit each of n-1 other cities once and once only and return to the start, as in Figure 4 17. The distance between each pair of cities is known. The task facing the salesperson is to organise the visits in such a way that the minimum distance is travelled

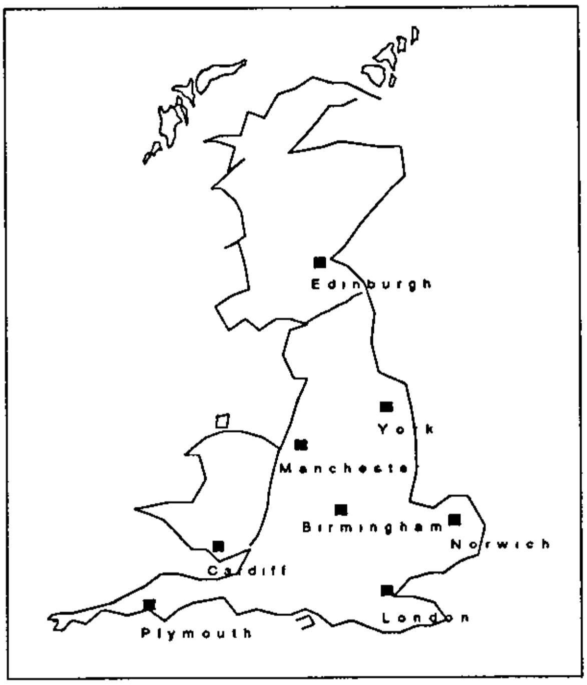


Figure 4.17 *Map of cities to be visited by travelling salesperson*

The problem is easy to state but difficult to solve, as for n cities there are (n-1)! possible tours. The number of possible tours grows exponentially in relation to the number of cities, as is shown in Table 4.2.

Cities (n)	(n-1)	(n-1)!
4	3	6
6	5	120
8	7	5,040
10	9	362,880
12	11	39,916,800

Table 4.2 *Number of Tours in TSP*

It is clear that to calculate every possible tour and to choose the shortest very quickly becomes a time-consuming procedure, and unrealistic for large values of n, and moreover values of n which are within the capacity of a single minibus A number of procedures have been used to attempt to reduce the computational requirements for this combinatorial problem. One of the earliest was the branch and bound solution proposed by Sweeney (1963) and Little et al. (1963). Bodin et al. (1983) survey both mathematical programming optimal techniques and heuristic techniques. In routeing problems large TSP matrices are rare, but the TSP can be applied to many problems other than those involving vehicle routeing, for example, finding the shortest path for a drill head drilling what may be hundreds of holes in a printed circuit board

To test the usefulness of these approaches, a version of the branch and bound algorithm was coded in PICK BASIC and tested with small TSP matrices. Even with small problems, the calculation of the solution was quite slow. This is because the PICK operating system and the BASIC language are designed for database handling rather than for intensive calculation, and because there are no matrix arithmetic facilities within BASIC. However, the most significant problem with the use of such algorithms to solve the routeing problems within the context of CT operations is the fact that the problem presents itself as that of finding an optimum tour between points on a plane rather than between nodes on a network. The location of each pick-up is known from its map reference; using the distances between pick-ups requires their calculation.

To calculate these distances requires one of two approaches. Either an approximate distance must be calculated between map reference grid centres using the same approach as that discussed above in the context of vehicle brokerage. Or a more accurate distance must be calculated using a detailed gazetteer and route-finding system in order to identify a measurable route between pick-ups. This latter approach is not realistic in the context of this system: detailed digital information on the road networks of towns and cities is not yet easily available, although the Ordnance Survey is in the process of digitising its maps. There are algorithms available which have been implemented to solve this kind of problem (Elliott and Lesk, 1982; Sugie et al, 1984) but these depend on the availability of detailed digital mapping data

A final approach, which is worth mentioning for the promise it may hold in the future is the use of neural networks These are electronic units or programs simulating neurons in the brain Hopfield and Tank (1986) claim that an electronic device built as a neural network could produce a good solution to a 300-city problem in about 1 microsecond

Given the inevitably approximate nature of the distance calculations, it is appropriate to consider other approaches to routeing which work on the plane rather than on a network It is also worth considering how humans order pick-ups in a route, for example experienced CT dial-a-bus drivers planning a route.

Derby CT runs a dial-a-bus service and also provides a group hire service using its own drivers for many voluntary organisations in the city. The approach used by Derby CT's full-time drivers, and which volunteer drivers are encouraged to use is one which uses a map and map pins

- A pin is stuck into a map of the city for each pick-up point.
- The driver then studies the map to identify a low-cost route. (Costs may include time as well as distance, and time may be influenced, for example, by avoiding right-hand turns onto main roads)
- The driver notes the pick-up order required for this route.

This description inevitably simplifies the process involved in the second stage. Waters (1984) has pointed out that humans are better than computers at recognising patterns for vehicle routeing and scheduling. The reason for this lies in the fact that a human, looking at a vehicle diary, a shift schedule or a map, is able to take in the whole, effectively at a glance, whereas a computer system must handle the data sequentially, considering each booking, each passenger trip or each pick-up in turn and comparing it with others in turn. A human uses a Gestalt approach to solving the problem, whereas a computer must use a sequential approach. Thus the driver may look at the map for a circuit which takes in most of the pick-ups, and then add outlying pick-ups into the route. He or she may also use heuristics, or rules of thumb, in order to decide where to start and finish, for example, the first pick-up will be the one nearest the vehicle base, and the last will be the one nearest the destination.

A technique for vehicle routeing which operates in the plane and which provides a similar approach to that used by humans has been described by Bartholdi et al. (1983a), Platzman and Bartholdi (1983) and Bartholdi and Platzman (1984). The technique is an heuristic based on the use of curves known as spacefilling curves, or Peano curves, and which are part of the family of fractal curves discussed by Mandelbrot (1983). The curves have a recursive structure, and constitute a mapping of the unit interval onto a higher dimensional space. The curve used by Bartholdi is shown in Figure 4.18, and its recursive structure can be seen. The more the unit square is divided into smaller grid squares, the more detail can be seen, the curve is infinitely crinkly. Any point in the unit square can be mapped to a point on the unit interval. From the point of view of its usefulness for routeing and other problems, the curve has an important feature: closeness in two dimensions is translated into closeness in one dimension. This is shown in Figure 4.19.

Bartholdi et al. describe a practical application of the curve to the routeing of meals on wheels deliveries (1983b), and the same approach can be applied to the ordering of pick-ups in dial-a-bus or group hire. The technique requires only that each grid coordinate is converted to a position on the unit interval, and that the pick-ups are then sorted by that value. This can be applied without the use of a computer, using look-up tables for the value of the position on the unit interval, the

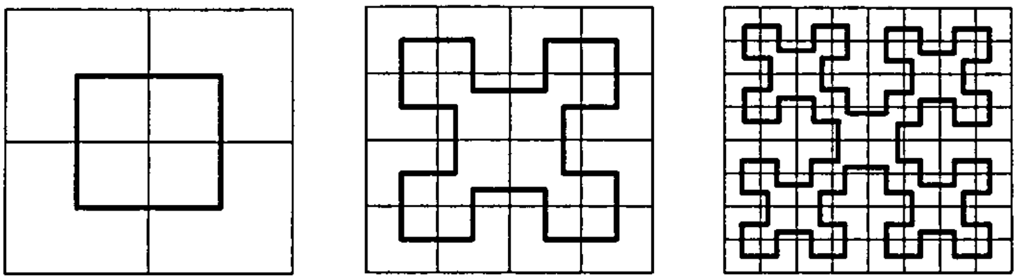


Figure 4.18 *Recursive spacefilling curve used by Bartholdi et al*

theta value Theta values were used in the first version of TRANSIT, as mentioned in Chapter 3, and subsequently dropped in a later version, presumably because dial-a-ride scheduling requires an ordering in time as well as space, and Bartholdi and Platzman suggest that it requires a mapping to five dimensions (1984)

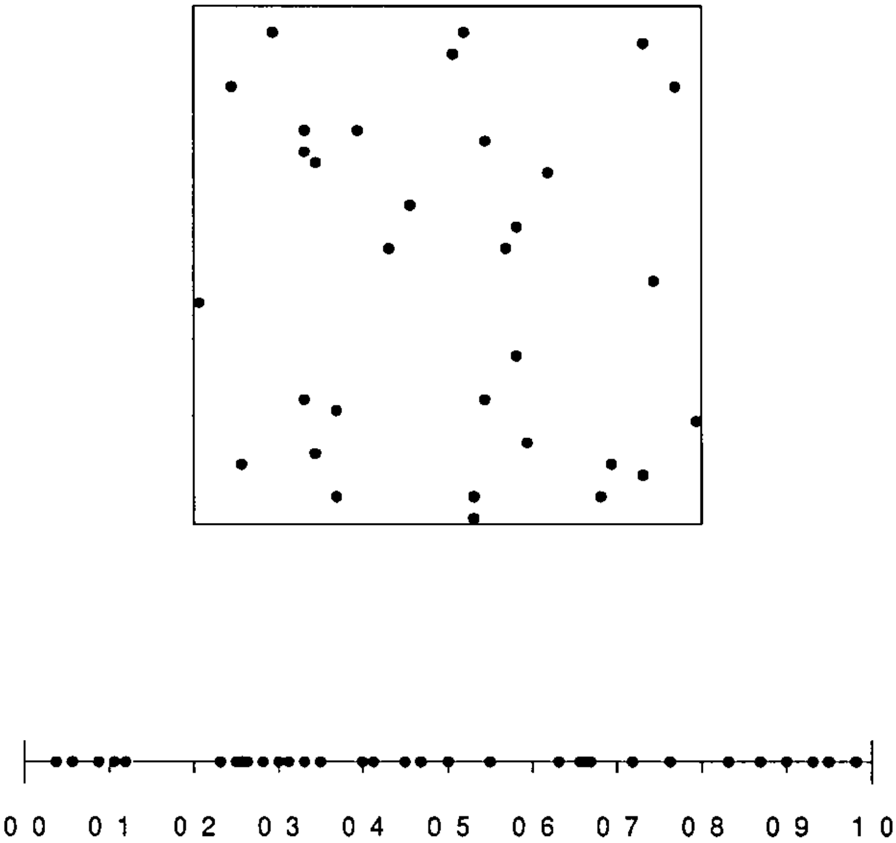


Figure 4 19 *Mapping of points in the unit square to points on the unit interval, preserving closeness and grouping*

In a computerised system, theta values can be precomputed for any point in a 100x100 grid square, and held in a file which is used as a look-up table. Using the PICK ACCESS enquiry language, it is possible to sort pick-up addresses by the theta value of their associated map reference expressed as x-y coordinates. The advantage of this approach is that it operates by consideration of n theta values for points on the plane rather than n^2 distances between points, and the computation required is only that of sorting rather than the combinatorial calculation required for the TSP.

The spacefilling curve approach to vehicle routeing was implemented in the software in order to organise group hire passenger lists It uses the sorting capabilities of the ACCESS enquiry language in conjunction with an F-correlative in the data dictionary for the passenger list file to provide a look-up on a table of theta values. It has been implemented in such a way that several passenger lists for a single booking can be merged together, sorted into a single tour and then manually partitioned into separate routes.

This approach does, however, have certain limitations.

- It only provides a first pass at the production of a suitable route
- It does not take account of natural and human-made barriers, such as rivers and railway lines with few crossing points, so points which are close together in the plane may not in fact be candidates for successive pick-ups.
- It does not take account of the limitations of passengers and their wheelchairs For example, it may be necessary that a passenger in an orthopaedic wheelchair is picked up last before traveling to the trip destination.
- It does not take account of any requirements for passengers to be picked up in a certain time sequence For example, a passenger who cannot be picked up until late in the tour as she is dependent on a nurse coming to dress her and prepare her to go out

These limitations will be explored further in Chapter 5 in the context of developing alternative approaches to vehicle routeing and scheduling

Dial-a-ride scheduling

The task of allocating passenger trips to dial-a-ride shifts, or dial-a-ride despatching is one that has been the subject of some study, both in terms of the manual procedures and as a problem in the field of operations research. Potter and Hagyard (1986) discuss and recommend a number of approaches to the task, including such aspects as layout of the despatching room and the despatchers' desks. There is no consistent approach to dial-a-ride despatching across the sector. A brief description of the methods in two dial-a-ride operators will serve to illustrate this. Descriptions of further approaches with one or more clerk or despatcher can be found in Potter and Hagyard (1986).

Derby CT operates a single 6-seater Renault Trafic as a dial-a-ride vehicle. The emphasis within Derby CT is on dial-a-bus for transport to the city centre and out of town shopping centres. The dial-a-ride service is for trips which cannot be met by the dial-a-bus service. The vehicle capacity is such that there is no great penalty in unused capacity if only a single passenger and companion travel. Many trips are made which are not multi-occupancy, although it is not infrequent that trips can be booked to overlap and improve the utilisation of the vehicle. The policy is that all outward trips must start, and all return trips finish, within the city boundary.

The procedure for taking bookings at Derby CT is that any member of staff will take a booking for any service, if he or she answers the telephone. For dial-a-ride bookings, a booking sheet is filled out for the outward trip, and one for the return trip. A check is made in the vehicle diary, shown as Figure 4.7 above, to decide whether the booking can be accepted. If accepted, the booking is written into the diary in the column for the dial-a-ride vehicle The staff member may negotiate the times of the trip or trips with the passenger in order to accommodate them. Exceptionally, and typically for regular users, a booking may be made onto one of the larger, 16-seater minibuses if it is not in use.

Barnsley Dial-a-Ride operates a larger fleet of vehicles, three of which are used for its dial-a-ride service. These vehicles are minibuses with a capacity of 12 passengers. The vehicles operate within the Barnsley Metropolitan Borough which includes several small towns to the east towards Doncaster, and villages on the edge of the Pennines to the west. Bookings are taken by booking clerks, who sit round a table equipped with telephones and with a card index of passengers, and each booking is written onto forms, one for the outward and one for the return trip. On the wall is a whiteboard divided horizontally by vehicles and vertically by times; each hour is divided into fifteen minute slots. Summary details of each passenger trip are written onto the whiteboard by the despatcher who stands by the board and who, using the information already on the board, a wall-map of the operating area and information from the booking clerks, seeks to allocate each trip to a suitable vehicle shift

In the case of Derby CT, because of the relatively small size of the vehicle, there is no strong requirement for multi-occupancy There is a limited resource available, and not much flexibility in how it is allocated; a deliberate policy has been adopted of trying to meet transport needs with a semi-scheduled dial-a-bus service.

In the case of Barnsley DAR, there is a greater resource available, and considerable demand for that resource, particularly at peak times of day. There is more flexibility in the way in which that resource is allocated. if trips are allocated efficiently, a significantly larger number of passengers can be carried. The despatcher is therefore always seeking to improve the vehicle utilisation, and as well as allocating new trip bookings to shifts, attempts to reallocate existing trips in order to improve efficiency The process of allocating trips to shifts is one which is heavily dependent on local knowledge, knowledge of the passengers and their disabilities, and common-sense While the task of allocating trips to shifts is the responsibility of the despatcher, the booking clerks also contribute, suggesting ways of resequencing or rescheduling trips in order to accommodate the maximum number of passengers, based on their knowledge of the area and the passengers.

While there are no formal targets, for example maximum times for which a passenger might expect to be on a vehicle, knowledge of passengers and their disabilities imposes constraints on the booking process, as some passengers may not be able to stand long travel times, or may need to be the last person onto and the first person off a vehicle because of the nature of their wheelchair.

In the afternoon, the clerks stop taking bookings, and prepare worksheets for the drivers for the next day, based on the information shown on the whiteboard and the booking forms

The ROMC approach to decision making is helpful in analysing the despatching process at Barnsley DAR. The whiteboard provides a representation of the problem space. Operations performed on the representation are those of adding a new trip booking to a shift and moving an existing trip booking between one shift and another. The card index used by the booking clerks, the wall-map and the booking forms act as memory aids. The formal procedures of taking a booking onto the forms, and the informal communication between booking clerks and despatcher provide the control mechanisms

The comments made in the section on the booking diary in relation to the Gestalt approach and to the information processing view of human problem solving also apply to this process. In particular, the whiteboard provides a summary of the situation which can be seen at a glance, while the booking forms provide all the details, but these can only be absorbed by reading sequentially through as many as a hundred forms

The allocation of passenger trips to vehicle shifts is one to which the techniques of OR have been applied, and Bodin et al make reference to a number of approaches to combined routeing and scheduling problems, which are characterised by task precedence and time window constraints (Bodin et al., 1983) Unlike the travelling salesperson problem, in which the sequence of stops is chosen in order simply to minimise a cost such as distance or travel time, dial-a-ride scheduling requires consideration of the fact that some stops, pick-ups, must be made before others, drop-offs, and that certain pick-ups and drop-offs must be carried out within time window constraints, in order for example to catch a train or to get to a doctor's appointment Moreover, unlike the vehicle routeing problem of which the dial-a-bus service or the collection of group hire passengers for delivery to a common destination are examples, and which are many-to-one problems, dial-a-ride scheduling is a many-to-many problem, with multiple pick-up and drop-off points. In their general introduction to the relevant section of their paper, Bodin et al. state that,

''All of these problems are NP-hard In most cases, the complications in these problems are such that exact algorithmic approaches based on mathematical programming formulations have not proven successful.'' (Bodin et al., p.151)

Bodin et al. distinguish between static and dynamic versions of the problem. In the static or subscriber version, called advance-request by Jaw et al. (1986), passengers book in advance, and a complete database of demand is available before any scheduling takes place. In the dynamic or demand-responsive version, passengers make bookings in real-time, and the allocation of passenger trips to vehicles depends on the current state of the system in terms of passengers already on vehicles and passengers waiting to be picked up It is important to note that in the UK, the situation is different from that in the USA and Canada. In North America, where local telephone calls are free of charge, it is usual for passengers to ring in and make a booking and not be told immediately whether their booking can be accepted. When all the bookings for a day have been collected, then the scheduling is performed, and the paratransit office rings passengers back to tell them the outcome. Although, this approach is common, it is not universally accepted, and Cutler and Harman (1984) discuss the desirability of performing scheduling interactively with the client on the telephone in order to eliminate ''time-consuming and embarrassing call-backs''

A further difference is that it is common to sub-contract trips which cannot be accommodated to other transport providers, typically taxi services. Authors such as Alfa (1986) and Kikuchi (1987) refer to trips which cannot be accommodated being dealt with in these ways An example of the dynamic approach in providing public transport to conventional passengers is the Rufbus service run in Friedrichshafen in West Germany, mentioned by Jaw et al (1986) and described in some detail by Sutton (1980), in which passengers call in for almost immediate service In this case, scheduling is provided by a mainframe computer.

In the UK, the situation falls between these two extremes. Passengers are told whether they can be provided with the service they require at the time that they ring in, but they are making a booking typically for the next day. Thus, while it may be possible to run a batch scheduling routine at the end of the day to reschedule trips for the following day, it is important that a decision can be made at the same time as the booking, while the passenger is still on the telephone In general, unallocated trips are passed on, where possible, to local social car schemes, which also take bookings in advance, and not to taxi services, except, so far as is known, in the case of Readibus, descnbed by Absolon and Bowlby (1989)

The computerised approaches to schedulıng of dıal-a-nde tnps are based ın the North Amencan expenence. Given that there is no optimal algorithm for the multıple vehicle, many-to-many dıal-a-ride problem with time windows, the research emphasis has been on heuristic approaches to batch schedulıng of the advance-request version of the problem. Jaw et al. (1986), Alfa (1986) and Kikuchi (1987) all take thıs approach. They require that trip requests are sorted in advance, typıcally in tıme sequence, a requirement which cannot be met in a demand-responsive system in which bookings must be dealt with in the random order ın which they come from passengers

An alternatıve approach is one suggested by the work of Knott (1988) and Waters (1990)· an artıfıcıal ıntellıgence approach using expert system techniques However, both are concemed wıth the schedulıng of vehicle delıveries in dıstributıon systems rather than wıth the complexities of dial-a-ride scheduling Knott has developed a practıcal applıcatıon for emergency relıef schedulıng ın Prolog, but it ıs a batch process ın whıch the requırements are known ın advance

The approach whıch has been chosen ın the research has been to provide the bookıng clerk or despatcher with a representatıon of the ınformatıon he or she requıres to make the decısıon about a partıcular trip booking, and the abılıty to perform the operatıons mentıoned above addıng a tnp booking to a shift and swappıng a tnp bookıng between shıfts A sımilar approach ıs taken by the DART software used in Vancouver (Chown, 1989), ın whıch the emphasis is on providing the despatcher wıth the ınformatıon he or she requıres rather than on automating the scheduling process The facılıtıes for dıal-a-nde schedulıng in MULTI TRIP are sımılar to those provıded by the vehıcle dıary and centre around the dıal-a-ride shift rather than the vehıcle day. The role and structure of the dıal-a-nde shıft ıs sımılar to that of the vehıcle dıary It reflects a number of poınts

- that shıfts are used by dıal-a-nde operators and regarded as a separate entity from the tnps whıch make them up,
- that data attnbutes such as date, start tıme, fınısh tıme, driver are associated with shıfts, although ın Fletchei's Dıal-a-Rıde some of these attnbutes are

held in trip records and some not at all; and

- that booking clerks and dispatchers need to be able to check the status of shifts at a glance, which in manual systems and hybrid systems is done using wall charts or clip-boards.

The logical data structure for shifts is shown in Figure 4.20. The physical data contents are shown in Table 4.3.

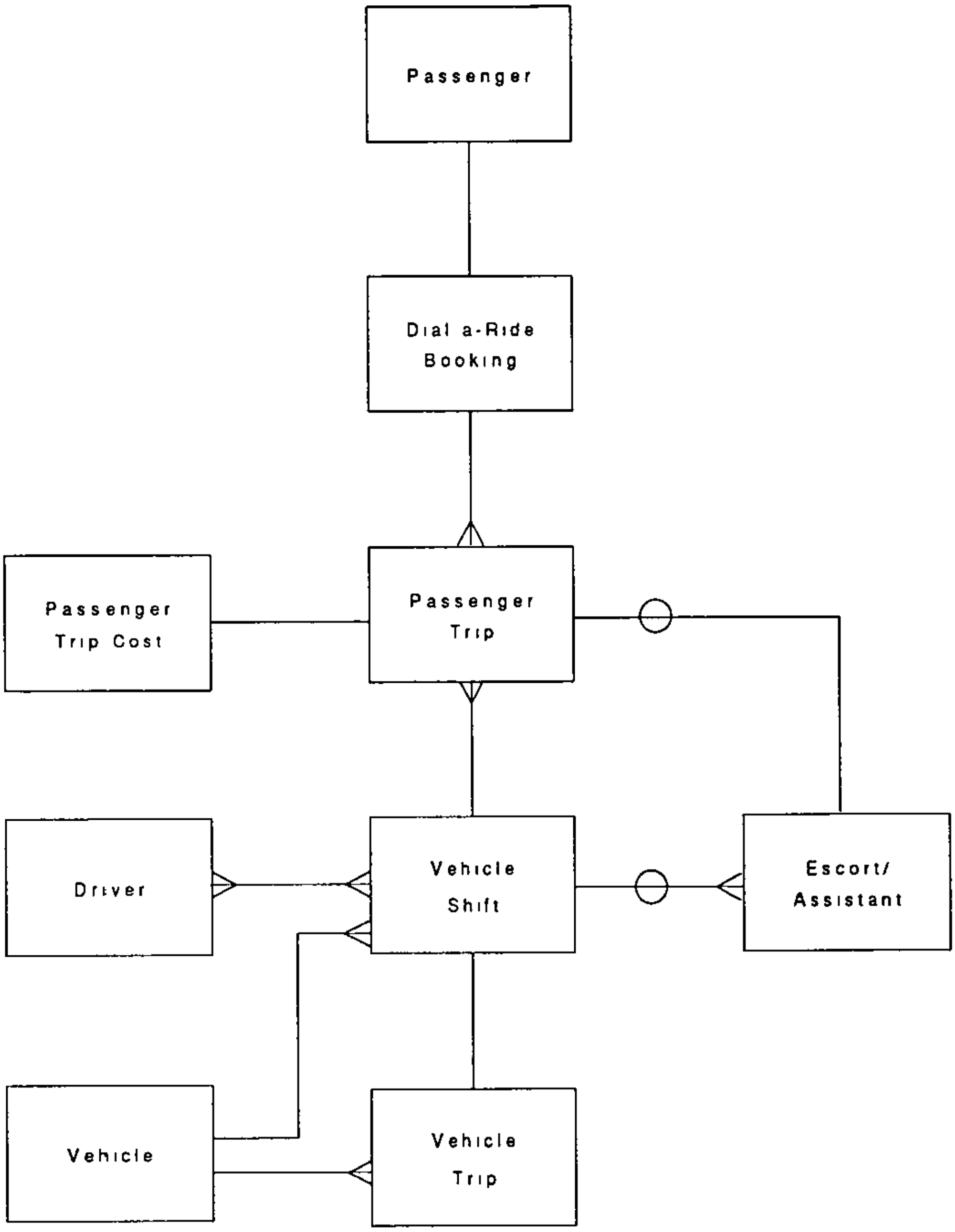
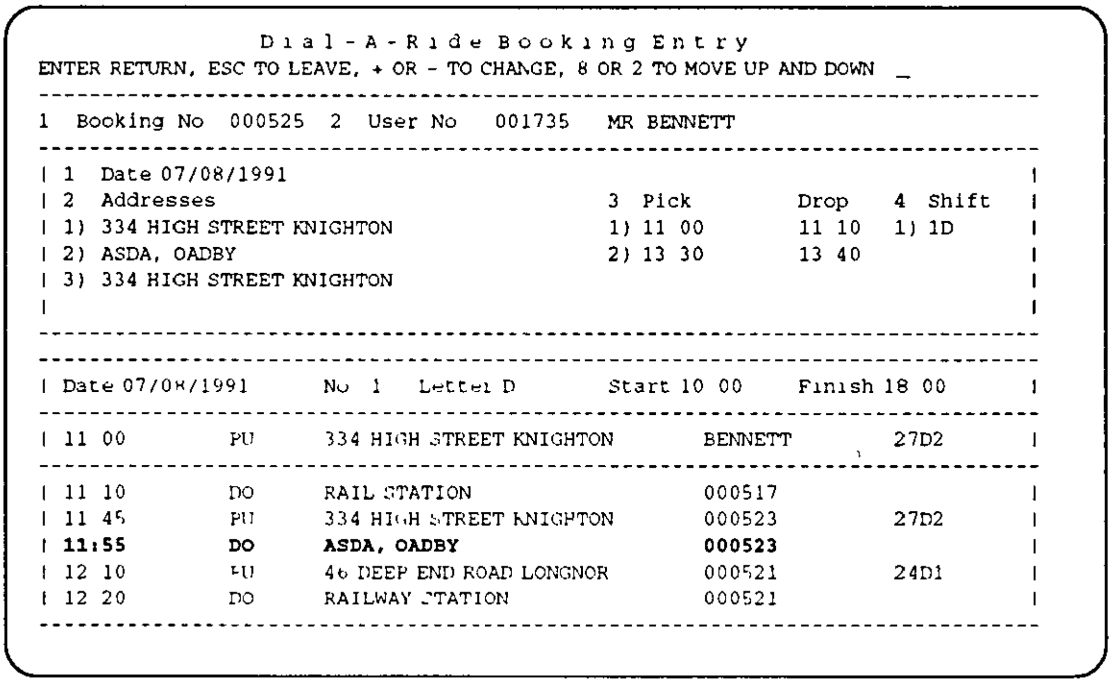


Figure 4 20 *Logical data structure for dial-a-ride shift*

8620/1/D				
Date	8620 07/08/1991			
Shift No	1			
Shift Letter	D			
Shift Start	36000 10 00			
Shift Finish	64800 18 00			
Trip No	8916691			
Vehicle No	0002			
Stop Time	36000 10 00	36600 10 10	36900 10 15	37500 10 25
Stop Type	PU	DO	PU	DO
Passenger	000519	000519	000525	000525
Address	98 LONG RD	RAILWAY STN	334 HIGH ST	ASDA
Map ref	23E1	23A3	27D2	24D1
Zone	D20	F14	E11	F12
Passengers	+0	-0	+0	-0
Wheelchairs	+1	-1	+1	-1
Num Map Ref	26,7	22,9	19,8	30,3
Booking Ref	0006890P	0006890D	0006955P	0006955D
Trip Type	O	O	O	O

Table 4 3 *Physical data contents of dial-a-ride shift*

As with the diary, this structure is heavily dependent on the implementation in PICK. A considerable amount of redundancy is introduced with the dial-a-ride shift. Much of the data held in the shift record is held elsewhere in trip booking records, but the system is designed in such a way that the shift is only ever updated by the process of making, amending or cancelling a dial-a-ride booking, which retains the integrity of the data. The redundancy is introduced to allow the data in a shift record to be displayed more efficiently, that is with fewer file accesses, than if the data were held in a number of separate files. This is required to allow the clerks and dispatchers to view shift records on screen while taking a booking, in order to ascertain whether a booking is feasible or not The quick booking screen for dial-a-ride booking entry is shown in Figure 4.21. The lower half of this screen shows the details of the shift under consideration for the trip which is being dispatched.



```
                    Dial-A-Ride Booking Entry
ENTER RETURN, ESC TO LEAVE, + OR - TO CHANGE, 8 OR 2 TO MOVE UP AND DOWN _
--------------------------------------------------------------------------------
1  Booking No  000525  2  User No   001735   MR BENNETT
--------------------------------------------------------------------------------
| 1  Date 07/08/1991                                                           |
| 2  Addresses                            3  Pick       Drop    4  Shift       |
| 1) 334 HIGH STREET KNIGHTON             1) 11 00      11 10   1) 1D          |
| 2) ASDA, OADBY                          2) 13 30      13 40                  |
| 3) 334 HIGH STREET KNIGHTON                                                  |
|                                                                              |
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
| Date 07/08/1991      No  1   Letter D       Start 10 00   Finish 18 00      |
--------------------------------------------------------------------------------
| 11 00        PU      334 HIGH STREET KNIGHTON      BENNETT        27D2       |
--------------------------------------------------------------------------------
| 11 10        DO      RAIL STATION                  000517                    |
| 11 45        PU      334 HIGH STREET KNIGHTON      000523         27D2       |
| 11:55        DO      ASDA, OADBY                   000523                    |
| 12 10        PU      46 DEEP END ROAD LONGNOR      000521         24D1       |
| 12 20        DO      RAILWAY STATION               000521                    |
--------------------------------------------------------------------------------
```

Figure 4 21 *Dial-a-ride quick booking entry screen*

The window in the lower half of the screen is divided into three sections.

The first section shows information about the shift which is under consideration, its start and finish times. The user can move to a different shift on the same day, using the numeric keypad. The shifts for the day are held in an array with, in effect, vehicles horizontally and shifts vertically. This information is held in a file which provides an index to shifts by day and in which the array structure is maintained. Thus by pressing the down-arrow or 2 on the numeric keypad, the user can move to a later shift on the same vehicle, say from an early shift to a middle shift, while up-arrow or 8 moves in the opposite direction. Pressing right-arrow or 6 moves to the next vehicle with a shift at the same time, for example, from vehicle 0001 early shift to vehicle 0003 early shift, while left-arrow or 4 moves in the opposite direction. Each time the user moves to a different shift, the data in the third section of the window is changed to reflect the shift under consideration. Finally, by pressing the key in the centre of the numeric keypad, 5, the user 'zooms in' to the next level, the second section of the window.

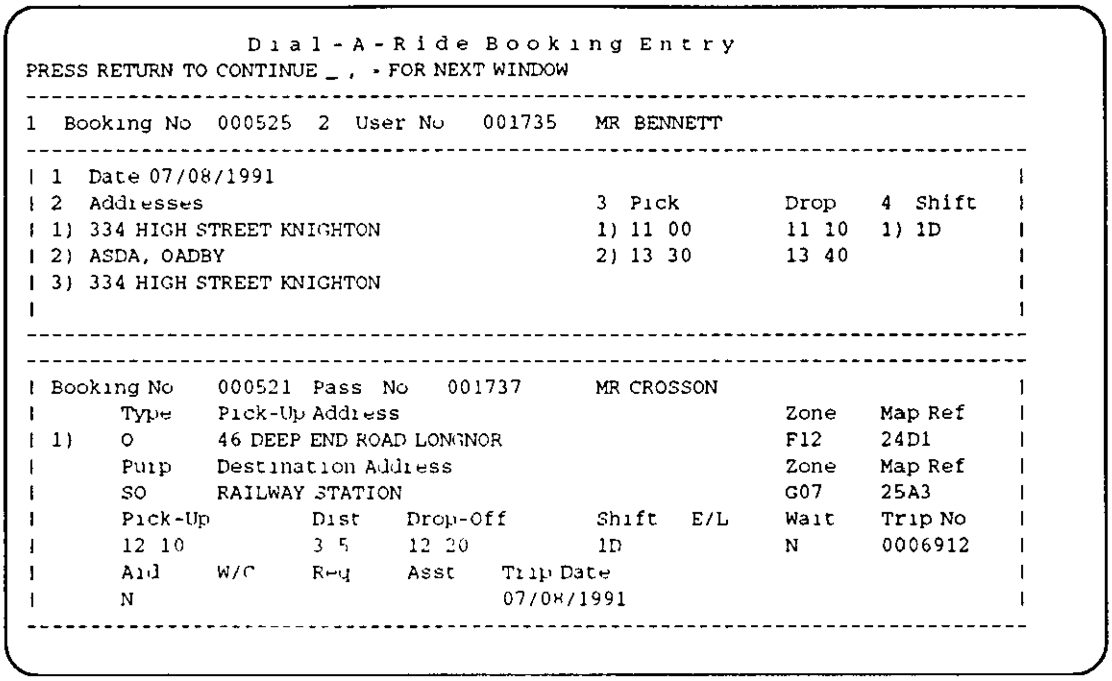
The second section shows information about the trip which is being despatched, the passenger and the pick-up or drop-off time and map reference details. Again using the numeric keypad, keys 2 and 8, the user can toggle between viewing the pick-up details and the drop-off details. In addition, the user can increment or decrement the pick-up or drop-off time in five minute intervals using the + and - (plus and minus) keys. It is not permitted to set the pick-up time the same as or later than the drop-off time. Again from this level, the user can press the 5 key to zoom in to the next level, the third section of the window.

The third section of the window displays summary details of the trip pick-ups and drop-offs already allocated to the shift displayed in the first section. When the user moves to the third section, one trip stop is highlighted (in bold in Figure 4.21) in the centre of the window. This trip stop will be the nearest in time to the pick-up or drop-off under consideration in the second section of the window The user can scroll the pick-ups and drop-offs in this window using the following keys

up-arrow/8 - up one trip stop
down-arrow/2 - down one stop
page-up/9 - up five stops
page-down/3 - down five stops
home/7 - beginning of shift
end/1 - end of shift

The final facility which is available to the user is to press the 5 key and to 'zoom in' and view in more detail the trip pick-up or drop-off currently under the highlighting cursor in the centre of the third section window. This is shown in Figure 4.22.

In this way, the user can view from within the booking screen, all the information necessary to decide whether and where a trip booking can be allocated. This provides similar facilities to the booking diary, in allowing the user both the checking function and the searching function in relation to the availability of space in dial-a-ride shifts The checking function is also provided in the software, when a trip is allocated to a shift, by checking that the trip does not clash with any existing trips for that shift.



```
                    Dial-A-Ride Booking Entry
PRESS RETURN TO CONTINUE _ ,  - FOR NEXT WINDOW
-----------------------------------------------------------------------
1  Booking No  000525  2  User No   001735   MR BENNETT
-----------------------------------------------------------------------
| 1  Date 07/08/1991                                                  |
| 2  Addresses                               3  Pick        Drop   4  Shift  |
| 1) 334 HIGH STREET KNIGHTON                1) 11 00       11 10  1) 1D     |
| 2) ASDA, OADBY                             2) 13 30       13 40           |
| 3) 334 HIGH STREET KNIGHTON                                         |
|                                                                     |
-----------------------------------------------------------------------
-----------------------------------------------------------------------
| Booking No    000521  Pass  No   001737    MR CROSSON                |
|      Type     Pick-Up Address                         Zone    Map Ref  |
| 1)   O        46 DEEP END ROAD LONGNOR               F12     24D1     |
|      Purp     Destination Address                     Zone    Map Ref  |
|      SO       RAILWAY STATION                         G07     25A3     |
|      Pick-Up          Dist    Drop-Off    Shift  E/L  Wait    Trip No  |
|      12 10            3 5     12 20       1D         N       0006912  |
|      Aid      W/C     Req     Asst    Trip Date                       |
|      N                                07/08/1991                     |
-----------------------------------------------------------------------
```

Figure 4 22 *Dial-a-ride quick booking screen, showing details of another booking*

The existence of the vehicle shift as a separate entity and thus as a file within the system has an operational impact on dial-a-rides. Barnsley DAR, which collaborated in the development of the software, operated a manual system based on booking sheets and a wall chart as described above Most bookings were taken during the morning and early afternoon, and the afternoon from about 3 00pm onwards was spent transcribing the details from the booking sheets and the wall chart to shift worksheets which could be given to the drivers. Birmingham Ring-and-Ride uses Fletcher's Dial-a-Ride software, which lacks any data structure for the shift, or even an index to link trips to a shift Bookings are taken on the computer, but shift records are still maintained manually on sheets of paper on clip-boards which can be passed around the despatchers The process of printing out computerised shift worksheets is a tedious one which involves searching and sorting the entire trips file, it is normally only run once at the end of the day in preparation for the next day. The impact of Fletcher's Dial-a-Ride on Barnsley would have been minimal While the printing out of the shift worksheets would not involve staff time in the same way as transcribing the booking sheets, it would still be a time-consuming process, and one which would limit the working day The use of the MULTI TRIP software at Barnsley has, however, meant that it is possible to print draft copies of the shift worksheets out at lunch time. Looking at these worksheets, an experienced despatcher can see where improvements can be made to the scheduling, and where passenger trips can be moved from one vehicle to another to save space and time. These changes can be implemented, and bookings can continue to be taken during the afternoon, with the possibility of repeating the process of printing draft worksheets until the end of the working day. The skills of the staff, who now combine the roles of booking clerk and despatcher are emphasised and developed; the computer system supports them in the decision making process rather than replacing them, and has been designed to reflect working practices in the CT sector in the UK

This chapter has examined the approach taken in computerising four operational functions in CT services based on an understanding of how it is that people perform those functions manually, and in particular of theories of human decision making. In one case, vehicle routeing, a technique from Operations Research has been used; in the others the computerised implementation is based on providing a tool for people to use which reflects the way they typically work, and which is based on the data model which underlies the system

In Chapter 5, conclusions about the software development are drawn, and suggestions for future developments are made, based on the data model outlined in Chapter 3, and using graphics and expert system technology to support the routeing and scheduling of trips

Chapter 5

Conclusions and Future Development

Introduction

This chapter aims to answer two questions which arise from the research and the development of the software. These are:

- What are the key features of the research, and what bearing do they have on the development of this and similar software?
- What further developments could be made to the software, and how can they be integrated with the developments to date?

The first question is addressed in the section which follows, while the second is addressed in the final section, which considers such developments in the areas of operations research, expert systems and graphical displays, and in scheduling architectures and understanding scheduling

Conclusions

Commercially available software packages for CT operations in both the UK and North America, as outlined in Chapter 1 and analysed in more detail in Chapter 3, have in the past been developed for a single operator and subsequently been taken to a wider market. In consequence, such packages have typically shown certain shortcomings.

- they cater for a single service;
- they encapsulate the working practices of the operator for which they were originally developed; and
- they have limited applicability to the sector as a whole.

In the realm of commercial software development, some kinds of packages are generally applicable to all kinds of businesses. Accounts packages, for example, must provide the same basic functions in order to record income and expenditure for a company. The differences between the requirements of a small business and those of a large multi-national are differences of scale and of complexity, for example multi-company and multi-currency, rather than of function. In the development of such systems, it suffices that the functions provided by the software should conform to standard practices. When software is developed for a single organisation, on the other hand, it is necessary only to analyse the requirements of that organisation.

The CT sector is diverse in the services that it provides and in the operational styles of different CT operators. For a single package to serve the needs of this diverse range of potential users, it must in some way reflect the operating practices of all such users In the MULTI TRIP package, this has been achieved in two ways: firstly, by analysing a wide range of operating practices through the collaborative development process and secondly, through extensive analysis of the underlying data model of such organisations, the enterprise model. The use of a decision support approach to the development of functions to support operational decision making is also of significance here

Collaborative development

The involvement of a number of CT operators in the development of the software formally both in the User Group and as Pilot Projects and informally through feedback on the software helped to ensure that the functionality of the software reflected operational practices across the sector. Software for vertical markets is often developed by software houses for an initial client and then extended to cater for a wider range of clients. In the process, customised versions are often produced for individual clients, and maintenance and support become increasingly difficult. This was the case with Fletcher's Dial-a-Ride, which was originally developed for Haringey Dial-a-Ride (Marsh and Jarrett, 1985), and then developed and customised for other dial-a-ride operators. Research shows that the cost of changing software increases the closer it is to implementation. It clearly makes sense to incorporate all requirements at the initial analysis stage rather than amending existing software.

Many software houses encourage the development of user groups for their software, and this typically occurs once the software has been developed and is in use by a number of clients. The user group provides a forum for discussion of problems with the software and for suggestions for future development. By channelling such suggestions through a user group, they can be filtered and corrected to represent a single view and avoid further customisation.

The involvement of the User Group and other forms of collaboration from the start of the development of this software have made it possible to develop the software to a single model which caters for a wide range of users from the outset. To some extent, this may be seen as a luxury only available to a research project which is not constrained by the commercial requirement to cover development costs. Yet the success of the software indicates that revenue from hardware and software sales, from maintenance contracts and from training makes the package commercially viable for a small company.

The suggestion that users should be involved in the design of software systems is one that is gaining support. The traditional systems life cycle approach isolates users from the development team. Users provide the material for the initial analysis of requirements, and sign off a statement of requirements At other stages, they are asked for feedback and to sign off subsequent analysis and design deliverables. This provides the developers with a semi-legalistic let-out when the software does not perform as required by the users: "They signed it off." Many analysts find the involvement of users a hindrance, as they change their requirements as the project develops.

However, research indicates that a high proportion of computer systems do not work as required, and that many are never used. Articles in the computer press, for example Davidson (1991), report recent studies which show that 'less than 1% of completed large software system projects are finished on time, within budget and meet all user requirements'. There are three possible approaches to the problem of the inability of system development processes to deliver what is required. The first is the project management approach, typified by Brooks (1975), stressing the need to organise the overall project and the development team in a way which is likely to improve the chances of success The second solution often proposed in order to solve this problem is a technical fix, software engineering, addressing the back-end rather than the front-end of the cycle, using mathematical techniques to specify processes and to transform them formally into 'correct' program code. The third approach is the one which has been adopted in this research Here the emphasis has been on the front-end, on getting the analysis right and involving users extensively in that process. It is not enough, however, simply to involve users in the development process. That involvement needs to be structured in a way which proves constructive both for the users and for the developer. In the research, the involvement of users has been structured in two ways: firstly, according to the level of involvement of any particular user or potential user organisation, outlined in Chapter 2, and secondly, in terms of the iterative process of developing the data model, outlined in Chapter 2 and described in detail in Chapter 3.

The structured involvement was important, as it would not have been possible to analyse the working practices of the many operators who were involved to some small degree in the same detail as the working practices of the Collaborative Projects were analysed. Figure 5 1 shows the difference in the detail in which different operators' practices were analysed. The scale is relative rather than based on any absolute measurable quantity, and reflects the fact that other operators were involved in small ways, commenting on screen layout at the Community Transport Association Annual Exhibition, sending in an example document or explaining what type of charging system they used, while the Collaborative Projects were involved in very detailed analysis of every document they used, and every manual function performed. However, many more other operators were involved compared to a few Collaborative Projects, which provided a wide spread of practices.

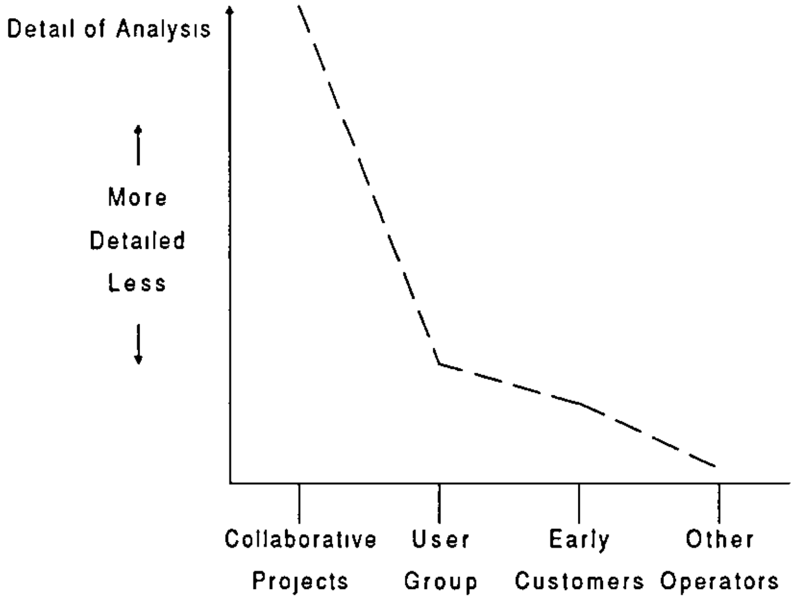


Figure 5 1 *Detail of analysis for different types of CT operators*

What this does not show is the difference in emphasis of the contact with the different types of operators. The early analysis focussed on what was common to the different operators in the User Group and in the Collaborative Projects. The later analysis focussed more on differences in working practices between different operators, including the early customers and other operators. This is shown in Figure 5.2.

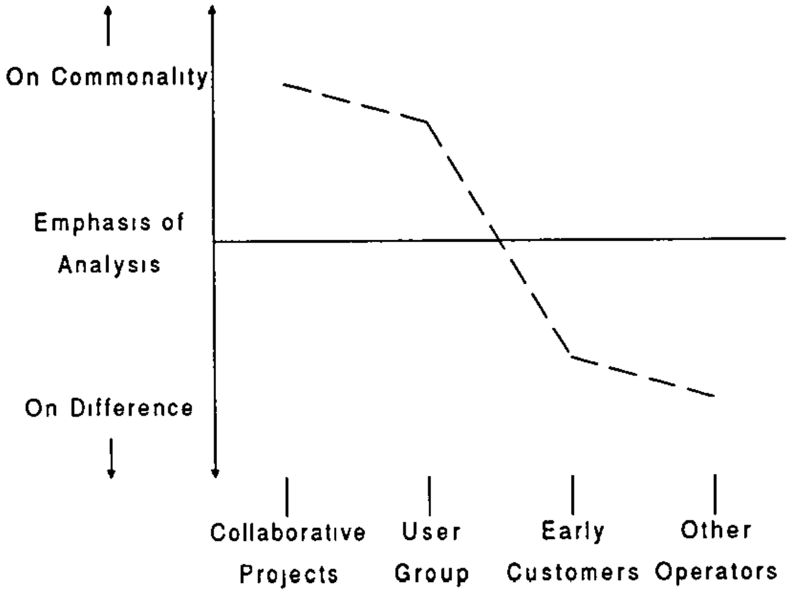


Figure 5.2 *Emphasis of analysis for different types of CT operators*

In this way, the majority of the data model and of the functionality of the software was defined through the analysis of what was common to different operators, while the refinements were made through analysis of what was different between different operators.

Underlying data model

It was the involvement of operators in the development which also ensured that the underlying data model reflected the requirements of the sector rather than of one or a few individual operators This process used the structure outlined in Chapter 2, by which the data model was successively refined, and which has been described in greater detail in Chapter 3 in relation to specific modules within the software.

It is important to recognise that the involvement of operators alone is not a criterion for success. The original User Group provided a list of files with proposed fields, referred to in Chapter 3 and listed in Table 3.2, at the start of this research. However, this list of files represented an attempt to move straight to a specification of what data should be held in a proposed computerised system, what Shlaer and Mellor call the 'Premature Rush to Implementation' (1988), rather than an analysis of what the logical data requirements of CT operators were. Such a logical analysis should be followed either by normalisation of the data, a bottom-up approach, or by entity-relationship analysis, a top-down approach, and only then to flexing of the logical model to a physical design.

The iterative process described in Chapter 2 led to the identification of entities and their attributes and the development of enterprise rules, which formed the basis for the entity-relationship modelling approach adopted here, and which represented the data requirements not just of a single organisation but of the sector as a whole. This has been described in Chapter 3. Of particular importance in this are features such as the clear distinction made between passenger trips and vehicle trips, and the recognition of the significance of the diary and of vehicle shifts.

Many authors since Chen (1976), have promoted the importance of data modelling, and of the entity-relationship approach in particular, as critical to the successful analysis of information system requirements Without adequate data analysis, it is difficult to provide the functionality required, as either the data stored in the system does not accurately represent the organisation and its requirements, or while the data is complete, it is structured in a way which means that it is unable to support the processing requirements of the system For example, Fletcher's Dial-a-Ride stores details of the vehicle and driver associated with every passenger trip. However, it stores this data in the passenger trip booking record. If normalisation techniques had been applied to this data, it would have been recognised firstly that it is not dependent on the passenger trip but on the vehicle shift, secondly that it is redundant, as the same information about vehicle and driver is stored in every passenger trip record for that shift, and thirdly that a vehicle shift should be a separate entity in the model of the enterprise. The result of this is that Fletcher's Dial-a-Ride is unable to support a process to display the current trips allocated to a shift except as a result of sorting and selecting from the entire passenger trip booking file. While the dial-a-ride shift in MULTI TRIP contains redundant data, this data has been introduced after normalısation, as a result of a positive file design decision and in order to minimıse the dısk accesses required to display the current status of a shift on demand.

The data model represents knowledge about the organisatıon and its information requirements, and in this case knowledge about many organisations and their requirements. This knowledge provides the foundation for the provısion of the operational functions of CT operation.

Decision support

The MULTI TRIP software package provides facilities for storing operational data and for extracting management information from that data. Much of that data relates to bookings for vehıcle and passenger trips. The recordıng of that data takes place in the context of operational decision making Staff taking bookings make decisions as they do so about how resources are to be allocated in order to meet the requirements of member groups and passengers Certaın functıons ın the software have been desıgned ın order to support thıs operatıonal decısıon makıng. These have been descnbed ın Chapter 4.

In order to provide these functıons, it was necessary to understand how staff ın CTs make these decisions, how they conceptualise the problem, how they use the informatıon that is avaılable to them and how that information is represented. Cognıtıve psychology and theory about problem-solving and decısıon makıng have been used In partıcular, Sprague and Carlson's ROMC approach (1982) has been used to provıde a model of the user's requırements whıch can be provıded by capabilıties within the software. Chapter 4 has described the way in which this was done for the four areas of the on-lıne vehicle diary, vehicle brokerage search, ordering of passenger pıck-ups and dıal-a-ride scheduling. The results are summarised here, and some general conclusions drawn.

On-lıne vehıcle diary

Manual dıary systems provide a view of vehicle utilısation 'at a glance', but further informatıon about vehicle bookings is usually held on booking forms which are not necessarily immediately accessible to the user. he or she may have to fetch them from another fıle. The diary system whıch has been designed in the software package described here provides the facility to check vehicle utilisation in summary format, but also to go straight from that summary format to the detailed information which in a manual system would be held separately. This is done in a way which provides three levels of detail: firstly, a very summarised view, which shows bookings simply as blocks (Figures 4.12 and 4.13), secondly, a view which shows exact times and details of the vehicle user (Figure 4.10), and finally, a view which shows the booking enquiry screen, with full details of the booking.

Diaries in general provide two functions: the checking function and the searching function The vehicle diary in the software provides the searching function in visual form to the user, while it provides the checking function in the form of a subroutine which checks the availability of any vehicle between specified times and dates.

The diary also allows the user to try out more complicated 'What if?' questions by allowing vehicle trips to be moved between vehicles or to be unallocated and held in the 'TBAL' list, while other adjustments are made.

Finally, the diary has been designed to allow the user to move between the diary screens and booking entry, amendment and enquiry screens and log sheet entry and amendment screens without having to go through the menu structure.

Vehicle brokerage search

The vehicle brokerage facilities within the software provide specifically for the search function required of a diary system. In brokerage the user has a booking to allocate to one of many vehicles, and as well as availability must consider the suitability of the vehicle to the needs of the passengers. Furthermore, it is important to ensure that vehicles are allocated in order of how difficult they are to book, and that overheads are minimised by allocating vehicles to bookings based on the distance of the vehicle base from the start point of the trip. These factors have been characterised in terms of the two types of criteria: availability criteria and sequencing criteria.

The software provides for these factors by using a sequence number and the distance of the vehicle base from the start point of a trip to order the vehicles for consideration, and by then checking each vehicle in turn for its availability and suitability. The resulting list and a list of unsuitable vehicles are displayed for the user, and he or she can select a vehicle from the former list.

Ordering of passenger pick-ups

The problem of sorting passenger pick-ups into an optimal or near-optimal order for drivers is an example of the travelling salesperson problem (TSP). In the traditional TSP, however, distances between points are known, while in the case of the passengers of a group hire or dial-a-bus service, the distances between points are not known, and would have to be calculated, but the locations of the points in the plane are known.

A version of the branch-and-bound approach to solving the TSP was coded and tested, but found to be slow. Moreover, it would require calculating the distance between each pair of passenger pick-up points. To do this accurately would require detailed data of the road network for the operational area and a route-finding algorithm, while if the distances are calculated based on the map reference grid, then they are inevitably estimates and their accuracy is dependent on the fineness of the map grid and on the nature and density of the road network.

For these reasons, a technique was adopted which works on locations in the plane rather than distances between points This is the space-filling curve heuristic developed by Bartholdi et al (1983), which associates a point on the unit interval with any point in the unit square using a function which preserves closeness and grouping of points. Pick-up points can then be sorted by a simple numeric value in order to provide a first pass route, which can form the basis for improvement by the driver's judgment

Dial-a-ride scheduling

Dial-a-ride scheduling requires similar facilities to those which are required from the diary, except that passenger trips which may overlap are to be allocated to vehicle shifts, whereas in the diary vehicle trips which must not overlap are to be allocated to vehicles

In the U.S.A., operational research techniques have been developed to provide batch scheduling programs which allocate bookings taken during the day to vehicle shifts for the following day, using call-back to inform customers whose trips requests cannot be met or are to be sub-contracted taxi services to provide those trips at a higher unit cost.

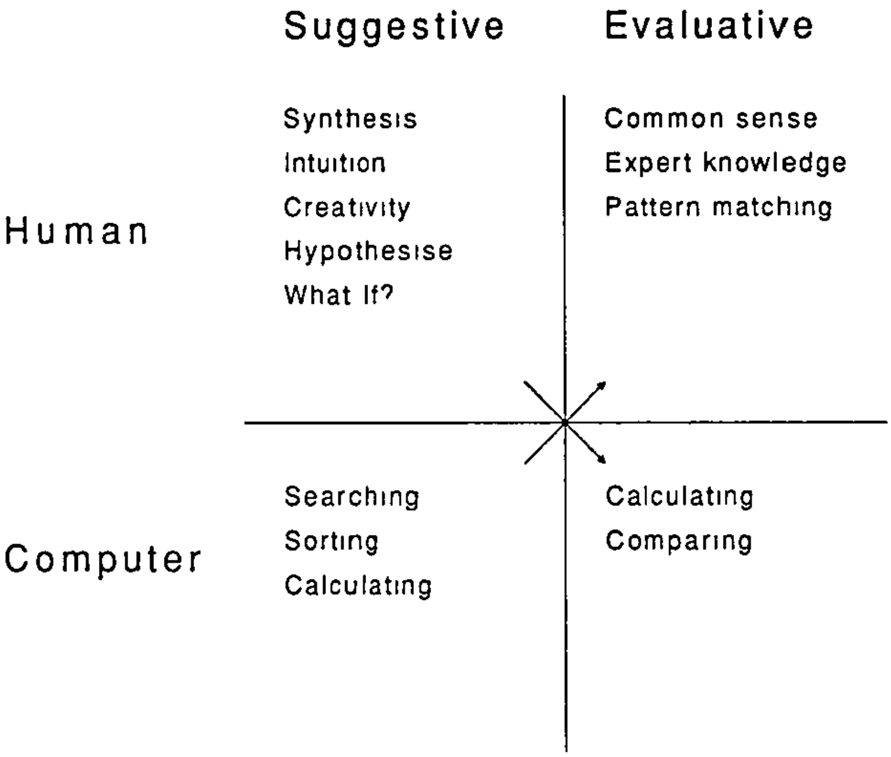
In manual systems in the U.K., whiteboards or clipboards with shift schedules on them act as a bottleneck in the despatching and scheduling process. These are designed to provide a summary of bookings taken, and the user still needs to refer to the booking sheet for each trip in order to get the full details, for example in order to consider reallocating a trip.

The software provides a facility similar to the checking function in the diary by checking whether a shift is suitable for a particular passenger trip. It also provides a facility similar to the search function by allowing the user to obtain a view of each shift at the time that the trip being despatched needs to be allocated, and allowing the user to move quickly from shift to shift.

In a similar way to the diary, the user can also highlight an existing allocated booking in a shift (Figure 4 21) and view the booking details for that trip (Figure 4.22)

General conclusions

There are two key points to be made about the way that these functions have been implemented in the software Firstly, these functions are designed to support operational decision making. They can be regarded to some extent as decision support systems, although that term is more often associated with systems designed to support unstructured or semi-structured problems of a strategic or tactical nature. DSS are usually considered in terms of factors such as the level of the decision (strategic, tactical, operational), the level of structure (unstructured, semi-structured, structured) or the elements of the DSS (dialogue, data, models). In the case of this research, a factor which is useful in analysing the support to human decision making is the relationship between the human user and the computer system. Two roles can be played by the human and the computer, suggestive and evaluative, and each may play either role. The human may ask a 'What if?' question, suggesting the allocation of a vehicle trip to a particular vehicle, and the computer may evaluate the suggestion, checking the availability and the suitability of the vehicle. Altenatively, the computer may search the database for a set of suitable vehicles for a brokerage trip and suggest them to the human, and the human may select one based on his or her judgment of which is most appropriate. This relationship can be seen in the functions which have been summarised above. However, the activities carried out by each, the human and the computer, when they assume each role are not the same. The activities and the terms we use to describe them clearly separate into those that are cognitive and require intelligence and those that are algorithmic and concerned with processing data. This is shown in Figure 5.3.



	Suggestive	Evaluative
Human	Synthesis Intuition Creativity Hypothesise What If?	Common sense Expert knowledge Pattern matching
Computer	Searching Sorting Calculating	Calculating Comparing

Figure 5.3 *Activities and terms associated with humans and computers in the suggestive and evaluative roles in the decision support relationship*

The software provides functions which perform both roles, both suggestive and evaluative where appropriate, and it is this notion of appropriateness which is the second key point People and computers are good at different things. People are good at tasks which require creativity and judgment or at pattern matching or simply at exercising common sense; computers are good at searching large quantities of data and performing repetitive calculations.

It is suggested, therefore, that the aim in software design should be to identify and understand those decision making tasks which people perform which are best performed by people and to use the computer system to provide the information required in order to perform those tasks, and to identify those tasks which are best performed by the computer and to ensure that the computer serves the requirements of the human user in performing them. In this way, decision making tasks can be partitioned between the human user and the computer.

Finally, it is useful to tie these conclusions into other recent research and other writings on the application of computer technology in business settings. Ainger (1991) reports the findings of an ESPRIT project (1217/1199) on the impact of computer technology on manufacturing, computer integrated manufacturing (CIM). He contrasts the early application of IT to manufacturing, which he characterises as the technological change approach, and which led to inflexible planning systems, with the more recent application of approaches such as just-in-time ordering and manufacturing systems, which he characterises as the organisational change approach Both are designed to meet the needs of high product quantity, low product variety production systems, and neither provides the necessary flexibility in today's low product quantity, high product variety environment The middle path proposed by Ainger is the human-centred approach, in which he suggests, 'the IT tools provided at both the cell level and the factory level are not solution generators but can be thought of as suggestion makers, enabling the personnel within the organisations to make decisions based on their local, accurate and up-to-date knowledge' (Ainger, 1991, p 60).

The approach taken has not been to automate and deskill, which is so often the outcome of computerisation, but in Zuboff's terms (1988) to 'informate' (inform/automate) and re-skill. This is appropriate in a sector which has its roots in community development and community action

Impact on CT operators

The effect of the software on CT operators provides some measure of its success. The fact that it is now the most widely used package for CT in the UK would appear to indicate that it meets the needs of operators. Continued contact with users reflects this, and two examples from users suggest that the software is accepted and meets management as well as operational requirements.

Barnsley Dial-a-Ride

Initially, Barnsley Dial-a-Ride retained its whiteboard for displaying trip allocations, as described in Chapter 4. However, it was recognised within a short time that the on-line shift scheduling capabilities provided the information that was required by booking clerks, and the whiteboard was removed. Bookings are now made entirely with computer support. An evaluation of the human computer interface design of the MULTI TRIP package (Ayub, 1992), and its use at Barnsley highlighted many problems and much potential for improvement, but drew out the fact that staff at Barnsley Dial-a-Ride had not considered changes to the software as it was, in their opinion, so much better than the old manual system.

Camden Community Transport

The last few years have seen tremendous growth at Camden CT, from a situation in which the organisation operated three vehicles and managed one other to a situation today in which Camden CT is responsible overall for the management of thirty-six vehicles within the borough. While this growth is clearly the result of good management and imaginative responses to opportunities which have presented themselves, Ed Passant, the co-ordinator indicates that the availability of management information provided by MULTI TRIP has been critical in assessing opportunities and presenting Camden CT's case in tendering for services

A report evaluating the software commented in relation to its applicability to commercial and public sector transport services, such as non-emergency ambulance, that ''features found in MULTI TRIP are probably unequalled elsewhere, especially the Diary'' (Browne, 1992).

Future Developments

A number of areas for future research and development are suggested by the present research. These can be seen to fall into two categories. The first concerns the 'How?', the second the 'What?'. Into the first category falls research into the methods by which software systems may be analysed, designed and developed; into the second category falls research into improving operational decision support.

Analysis methods

The MULTI TRIP package has been developed in collaboration with a number of CT operators. The approach adopted may be applicable to the analysis, design and development of other software with a heterogeneous user base. From the point of view of systems analysis and design, a more methodical exploration of the impact of the involvement of users in this process would be of use in establishing whether user involvement helps or hinders the development of a system and whether specific approaches can be adopted which promote beneficial rather than detrimental involvement.

Also in this first category, is the issue of developing human-centred systems. In systems which provide operational decision support, techniques for analysing tasks and determining whether the role of the computer system is to be suggestive or evaluative and how tasks are to be partitioned between the human and the computer would be of use. In particular, it is necessary to address the question of assessing the cost and the utility of attempting to develop computer systems in which the computer is intended to perform some of those functions which people are good at. As Ainger says 'People after all have the characteristics which we strive to emulate on our most sophisticated computer systems People are also very much cheaper and easier to support than many computer systems.' (Ainger, 1991, P 59)

If we view the human user or users and the computer or computers as a single system, and that system has a decision making task to perform, then it is possible to assign parts of that task to the human and parts to the computer. The cost of assigning different proportions to each will vary: if we assign parts of the task which involve performing repetitive calculations to the human, then the process will take longer, and wage costs will be higher; if we assign parts of the task involving judgment to the computer, then the software will have to incorporate artificial intelligence techniques and will be more expensive and may well require more expensive hardware to run it In designing the system, the task is to identify the point on the cost curve where the overall cost is lowest A hypothetical curve is shown in Figure 5.4.

For straightforward data processing tasks and for tasks involving the production of management information, the point at which the line is drawn between the human

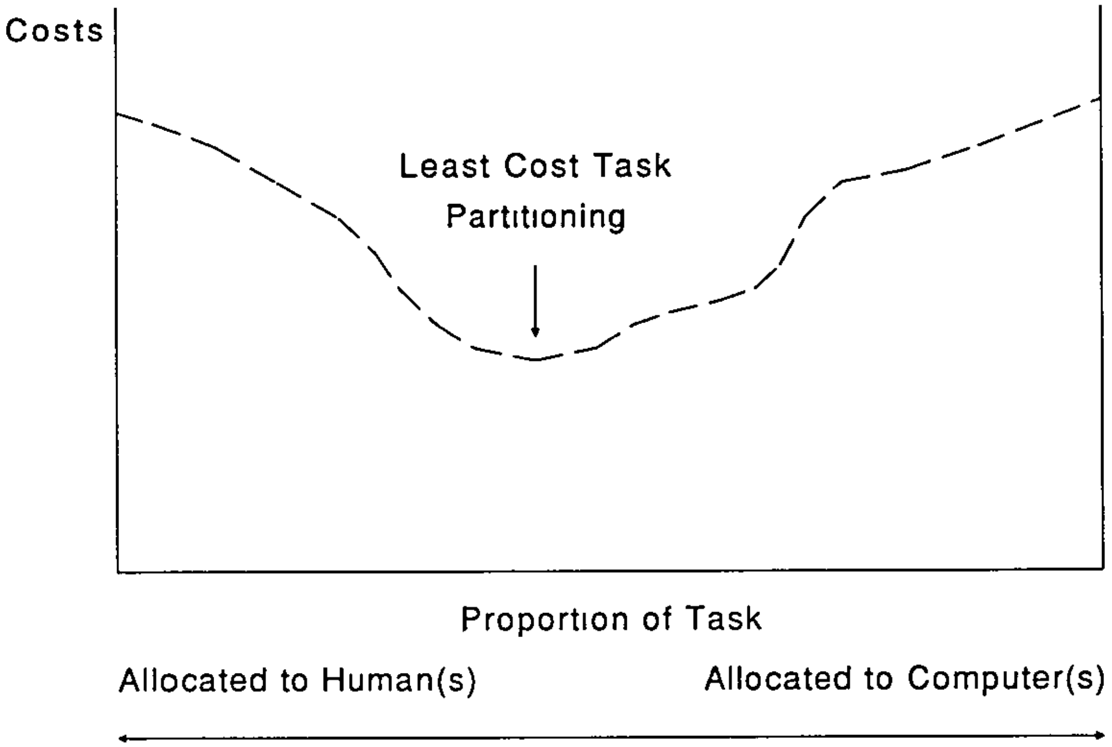


Fıgure 5.4 *Graph of system costs agaınst proportıon of decision makıng task allocated eıther to human(s) or to computeı(s)*

and the computer is usually easy to define. There may be options wıth different costs associated with each, but these are often to do wıth the choice of technology for the interface, for example keying data in compared with the use of bar-codes. For tasks involving decısıon makıng, determınıng the optimum point ıs lıkely to be more complıcated. Moreover, ıf we add a third dimensıon of functionalıty, then the optimum point is lıkely to move, potentıally ın either dırection, as the functionality of the overall system ıs ıncreased

For decısıon making systems, determining the costs is also lıkely to be more dıfficult than for data processing or management ınformation systems, as the total system cost ıs the sum of the human system cost and of the computer system cost, and comparing the cost of a human expert wıth an expert system package or a sophisticated graphıcs system may not be as straightforward as comparıng the cost of four staff with keyboards and one member of staff wıth a laser bar-code reader.

It ıs suggested that further research into the use of operational decision support systems, the way in whıch the decısıon makıng task has been partıtioned and the cost-effectıveness of that partıtıonıng would be of use.

Operational decision support

The second category of areas for potential research concerns what is to be developed rather than how it is to be developed or how design decisions are to be made. In the context of what has been said already, there are a number of ways in which software to enhance the operational decision making tasks could be developed, based on further research into the computerisation of functions within CT operations.

Four of the areas which are discussed below represent developments in the application of computerised technology to CT operations, and in particular to dial-a-ride and other passenger rather than group based services. The final area discussed concerns more fundamental research into the way that people schedule dial-a-ride trips, an understanding of which underlies the other areas.

Operations research

The approach adopted in the development of the software has been to understand the information requirements of operational decision making and to provide information in a way which supports that decision making In the context of the ROMC approach, the emphasis has been on the representations of the information rather than on the operations. The emphasis of OR is rather on the operations that are to be performed, either algorithmically or as heuristics. The support function of the software could be extended by applying further techniques from OR to the process of allocating resources to bookings

Most research in this area to date has been in developing approaches to batch scheduling in advance request dial-a-rides. There is little material on approaches to evaluating alternative possible allocations dynamically in demand-responsive dial-a-rides. Two approaches suggest themselves In the first, when a booking is taken, a scheduling function evaluates the possible options in allocating that booking to a shift and suggests a solution to the member of staff taking the booking. The staff member can then accept the solution proposed or reject it and ask for an alternative or propose his or her own In the second, the member of staff suggests an allocation and the scheduling function evaluates that allocation in terms of the cost of that to the organisation, the passenger and to other passengers already allocated to that shift. An approach such as Kikuchi's (1987), which allocates costs in terms of the disutility to passengers of wait times and excessive time spent on the vehicle and disutility to the organisation in terms of low vehicle capacity, could be applied to this situation. One of the problems with scheduling of this sort is the probabilistic and inexact nature of many of the variables. Times and distances are estimates and may be affected by traffic conditions, weather, vehicle problems and passenger delays. Kinkuchi's recent work with others (Kikuchi and Donnelly, 1992; Kagaya, Kikuchi and Donnelly, 1991) on the application of fuzzy set theory to such problems provides a promising route towards the development of scheduling algorithms and heuristics which can cope with such inexact data. It may also be that the use of fuzzy set theory provides a means of using numerical approaches to deal with some of the problems involved in scheduling passengers with complex needs, which it has been suggested can only be dealt with using symbolic logic and reasoning and which therefore fall into the realm of knowledge based systems.

Knowledge based systems

It has been commented above (Chapter 4) that the process of scheduling dial-a-ride trips is not simply one of numbers but is dependent on knowledge of the passengers and their requirements and of the operating area Artificial intelligence techniques may be able to offer assistance in automating the application of such knowledge. Two possibilities are suggested

Firstly, knowledge about passengers could be embedded in the software through the use of an expert system approach. While expert systems may be inappropriate to problems involving temporal or spatial knowledge as suggested by Yazdani, they may be applicable to knowledge about passengers and their special transport requirements Knowledge about passengers and their limitations to mobility, about the feasibility of travel from one zone to another may be ways of addressing the shortcomings of the space-filling curve approach to routeing which were outlined in Chapter 4.

Secondly, artificial intelligence techniques in the field of machine learning could be applied. While it may be difficult, if not impossible, *a priori* to express a set of rules which represent the knowledge used by a human despatcher in allocating trips to shifts, it may be possıble to develop a knowledge base which is based ın experiential learning and feedback. In particular data about actual travel times and distances could be entered into the system after trips and used to refine the system's understandıng of how to estimate travel times and dıstances between zones at different times of day.

However, ıt should be pointed out that research in this area largely contradıcts the suggestion that has been made earlıer that the emphasis should be on computers doing what computers are good at and humans doing what humans are good at. The approach taken so far has focussed on the presentation of appropriate information in an appropriate format to the human user, in order to suggest solutıons. In that context, for routeing and schedulıng problems, the graphical presentatıon of informatıon ın map form ıs clearly an approach which reflects closely the way ın which people work at present.

Graphics

In solvıng problems as outlined ın Chapter 4, humans make extensive use of geographical information, both in the form of local knowledge which ıs held in long-term memory and ın the form of maps which represent the operatıonal area. The text-only termınals normally supported as dısplay devıces by the PICK operating system orıgınally made the consıderation of the display of geographical ınformatıon in map form unrealıstıc However, as a representatıon of the information about the locatıon of vehicles and passengers, maps are prımary in manual systems.

Interest in and technıques for geographical ınformation systems (GIS) have developed rapıdly recently, and the ESRC supports regional research centres for the application of GIS technology It is possible to integrate PCs as terminals into PICK systems, and to use PCs to dısplay bıt-mapped or vector graphics. At Camden CT, attempts have already been made to use an off the shelf graphıcal routeing package, Autoroute, to dısplay routes graphıcally and to cost long distance group hire trips using it. The possıbılıty of usıng geographical data and representing it graphıcally on PCs linked to the PICK database is worthy of ınvestigation. At least three possıble methods for doing this exist. Fırstly, a number of terminal emulatıon packages whıch run on PCs provıde a vector graphıcs dısplay emulation, typıcally

TEKTRONIX 4014. Control sequences sent from the PICK machine can draw vectors on the PC screen. This is slow, as each vector must be sent as a series of codes on the serial connection, but it has the advantage of centralising the map database on the PICK machine. Secondly, a standalone mapping package could be run on the PC which stores the map data locally to the PC; data could be written to a DOS file by the PICK system before it invokes the mapping package which could write data to a file to be read subsequently back to the PICK system. This approach could be adopted with a package like Autoroute and a communications package such as VIA-Duct which allows the PICK host to control processes on the PC (VIA Systems, 1991). Finally, a graphical package could be developed which would communicate with the PICK machine The PICK machine would act as a database server providing data to the program on the PC, but the graphics would be drawn on the PC and the interaction between the user and the system would be handled by the PC. Each of these approaches seems feasible, but the technical and economic feasibility of each needs further evaluation.

Scheduling architectures

Three suggestions for enhancing the scheduling functionality of the software have been outlined above. These do not address the issue of how the functions are to be distributed. As it stands, MULTI TRIP does not distinguish between the booking and despatching functions. The distinction in the old system at Barnsley between booking clerks and despatchers has been removed by providing the booking clerk with all the information required to allocate the trip. This may not be the most efficient way of allocating the functionality. If other scheduling approaches are adopted, consideration must be given to alternative architectures.

These possible architectures are based on two dimensions:

centralised - distributed;

automated - supporting

The distinction between centralised and distributed functions is based on the approach previously used in Barnsley, which represents a centralised manual system The scheduling or despatching function is performed by one person, while the bookings are taken by more than one other people The current architecture in

MULTI TRIP is distnbuted in that the despatching funtion is performed by each user of the dial-a-ride booking programs. The distinction between automated and supportmg processes is that between programs which produce a schedule without the intervention of a human despatcher and those that work in conjunction with a user either suggesting or evaluating solutions as described in the general conclusions and operations research sections above.

Of the three suggestions made above for enhancing the scheduling capabilities of MULTI TRIP, only two lend themselves to implementation as automated processes, these are the OR approach and the expert system approach. By its nature, the use of graphical displays requires the involvement of a human despatcher. Four examples are descnbed here in order to clanfy how such architectures might work in practice

In a distnbuted, supportmg system usmg a graphical display, the user of the schedulmg software would enter the requirements for a passenger trip, including the map references and the times for the start and finish of each tnp. Having entered these, a map of the area would be displayed, either on the same screen or on a neighbouring screen, showing the location of the passenger trip start and finish and the stopping points of vehicles in the area at around those times. The user would be able to select a vehicle to which to allocate the tnp by pointing at it with a mouse cursor or possibly using a touch screen

In a distributed, automated system using a scheduling heunstic, the user of the schedulmg software would enter the requirements for the passenger tnp. The schedulmg software would be run as a called subroutine and would produce an allocation for the tnp based on evaluating cost functions for inserting the tnp into each possible shift

In a centralised, supporting system based on a graphical display, the user would enter the requirements, these would be stored on disk and passed into a queue. At a separate terminal, the despatcher would be presented with each tnp requirement in turn, and would view the trip start and finish and vehicle stops on a graphical map display. The despatcher would allocate each tnp to a vehicle shift, and the result would be stored and possibly also passed back to the booking clerk.

In a centralised, automated system using a scheduling heuristic, the user would enter the requirements which would be stored and queued A program runnmg as a separate process, without necessarily having a terminal attached to it, would take each trip from the queue in turn and allocate it to a vehicle shift and store the result and optionally pass it back to the booking clerk.

The reasons for implementing a centralised system as opposed to a distributed one might be based on the cost of resources such as graphical display screens or on the need in large multi-user systems to prevent lock-outs when several users attempt to update the same shift record at the same time.

Understanding scheduling

Central to all of these issues is the need to understand how it is that people make scheduling decisions. What heuristics, or rules of thumb, do people use to allocate trips? What knowledge do people bring to bear on the problem? How can information best be represented graphically? What are the organisational and communicational structures which best support the scheduling function? Underlying such questions are issues in cognitive psychology

Scheduling can be likened to solving a jigsaw puzzle, but a jigsaw puzzle in several dimensions. A trip must be slotted into place according to the dimensions of start location, finish location, start time, finish time and vehicle capacity. It would appear that people schedule trips by a Gestalt process similar to that by which they do jigsaw puzzles. In doing a jigsaw puzzle, one is provided cues in the form of shape and of pattern. It is necessary to ask whether, when scheduling, similar cues are provided, and whether analogies can be made, for example whether the passenger's requirements for a certain type of vehicle equate to pattern while the start and finish times and locations equate to shape. People can and do solve problems and puzzles in two and three dimensions; packing the car boot to go on holiday is a three-dimensional example as are wooden block puzzles. This raises the question of whether this is an innate ability in all people or one which is learned and can be improved with practice If it can be learned, can the components of the skill be extracted and documented in a way which would allow them to be implemented in a computerised system using artificial intelligence techniques? Research by Heim and Watts (1957) into the effects of practice and coaching on performance in tests of spatial awareness suggest that the skill can be improved in so far as psychometric tests are an accurate test of abilıty ın real world tasks, and indeed Heim (1970) states that a correlatıon between good spatial perceptıon and good practical abilıty has been repeatedly confirmed. If this is the case, then ıt may well be that some people make better despatchers than others, and that the despatching function is better separated from the booking functıon. Further research ın this field could contribute to improvıng the tools and organısational framework for dıal-a-ride scheduling.

The intention ın this last chapter has been to hıghlight features of the research which it ıs belıeved are sıgnıfıcant. The main product of the research has been the software package itself, which cannot be bound and submitted, but a recent summary of the system is ıncluded as Appendix A. The features referred to are fırstly, that ınvolvement of users ın the development of the software has contributed greatly to ıts success in terms of meeting users' needs, secondly, that through thıs user involvement a data model which embodıes the information requirements of CT has been developed whıch underlıes and supports the functıonalıty of the software and is critıcal to ıts usefulness, and thırdly, that taking a decısion support approach to specıfic operational decısıon makıng functions, and ın partıcular the ROMC approach and the partıtıonıng of tasks between the human user and the computer, has enabled the development of software which supports human skill in solvıng operational problems rather than deskillıng people

Further potential for research has been suggested ın two general categories The fırst relates to methods for analysıng and desıgnıng systems which involve users heavıly ın the development process and to determınıng the optımum balance between human and computer in decısıon making systems. The second relates to issues ın the applıcatıon of computerısed technıques to the operational problems which face CT operators Fıve areas for potential future research and development which build on the work undertaken here are suggested. These are in the use of OR technıques, knowledge based systems and graphıcal displays to support operatıonal decısıon making, ın the system architecture to support the use of these techniques and ın the development of greater understandıng of the cognitive abılıties whıch underly the task of despatching in dıal-a-rıde and sımılar CT operatıons.

Bibliography

P. Absolon and Bowlby, S., 'How to fill up Your Bus: The Readibus Experience', Fifth International Conference on Mobility and Transport for Elderly and Handicapped Persons, Stockholm, 1989.

A. W. S. Ainger, 'Manufacturing. a practical human-centred perspective', Engineering Management Journal, Vol.1, No.2, pp.57-62, 1991.

A. Al-Zobaidie and Grimson, J B , 'Expert systems and database systems: how can they serve each other?' Expert Systems, Vol.4, No.1, February 1987.

A. S. Alfa, 'Scheduling of vehicles for transportation of elderly', Transportation Planning and Technology, Vol.11, No.3, pp.203-212, 1986.

R. N. Anthony, 'Planning and Control Systems: A Framework for Analysis', Harvard University Press, Cambridge, Massachusetts, 1965.

N J Ashford, Bell, W. G. and Rich, T. A., 'Mobility and Transport for Elderly and Handicapped Persons', Transportation Studies Volume 2, Gordon and Breach, 1982.

Atech System Builder Ltd , 'System Builder Reference Manual', (printed from disk), 1986.

Atech System Builder Ltd , 'System Builder User Guide', 2nd Ed , 1986.

M. Avery, 'The Road Transport Industry's Guide to Software', Francis Pinter, London, 1985.

D. E. Avison, 'Information Systems Development: A Data Base Approach', Blackwell Scientific Publications, Oxford, 1985.

M. Ayub, 'Evaluation of MULTI TRIP', unpublished M Sc. Dissertation, Loughborough University of Technology, 1992

K Bainbridge (ed.), 'Vehicle Pooling Schemes: Seminar papers from a one-day training session organised by TRANSNET and the LCTA', TRANSNET, London, 1987.

J. J. Bartholdi, Platzman, L. K., Collins, R. L. and Warden W. H, 'How to Implement a Simple Routing System', Georgia Institute of Technology, School of Industrial Engineering, Atlanta, Georgia, April 1983.

J. J. Bartholdi, Platzman, L. K., Collins, R. L. and Warden W. H., 'A Minimal Technology Routing System for Meals on Wheels', Interfaces, Vol.13, pp.1-8, 3 June 1983.

J. J. Bartholdi and Platzman, L. K., 'Heuristics based on space-filling curves for combinatorial problems in the plane', Georgia Institute of Technology, School of Industrial Engineering, Atlanta, Georgia, September 1984.

J. S. Bate and Wyatt, M., 'The PICK Operating System', Collins, London, 1986.

P. Beecham, 'Computer at Birmingham Community Transport - A Report', Internal Report, December 1986.

P Belcher and Catling, I., 'Electronic route guidance by AUTOGUIDE: the London Demonstration', Traffic Engineering and Control, Vol 28, No.11, pp.586-592, November 1987.

P. C. Bell, 'Visual interactive modelling: The past, the present, and the prospects', European Journal of Operational Research, Vol 54, pp 274-286, 1991

W. G. Bell and Ashford, N. J. (eds), 'Third International Conference on Mobility and Transport of Elderly and Handicapped Persons', US Department of Transportation, Washington DC, USA, 1984.

S. C. Bennett, 'Designing a modular software package dedicated to Ct group hire, dial-a-ride and transport brokerage', in J C. Sutton with Gillingwater, D. (eds), 'Computer Applications in Community Transport - Which Way Forward?', Department of Transport Technology, University of Technology, Loughborough, Report No TT8703, July 1987.

S. C. Bennett, 'The Development and Implementation of Information Technology Applications in Community Transport: A Case Study of the MULTI TRIP Integrated, Modular Database Management System', in 'Mobility and Transport for Elderly and Disabled Persons', Proceedings of the 5th International Conference on Mobility and Transport for Elderly and Disabled Persons, Stockholm, Sweden, May 1989, Gordon and Breach, 1991.

S. C. Bennett and Gillingwater, D., ‘Systems Analysis and design Considerations in the Computerisation of Specialised Transport’, Specialised Transportation Planning and Practice, Vol 3, pp 115-137, 1989.

H. Bidgoli, ‘Decision Support Systems: Principles and Practice’, West Publishing Company, St. Paul, Minnesota, USA, 1989.

J. Bingham, ’Mastering Data Processing’, Macmillan Education, London, 1983.

L. Bodin, Golden, B., Assad, A. and Ball, M., ‘Routing and Scheduling of Vehicles and Crews: The State of the Art’, Computers and Operations Research, Vol.10, No.2, Special Issue, 1983.

P W. Bonsall and Kirby, H. R., ‘The role of expert systems in transport’, in P. W Bonsall and Bell, M (eds), ‘Information Technology Applications in Transpiort’, VNU Science Press, 1986.

R. J. Bourdon, ‘The PICK Operating System - A Practical Guide’, Addison-Wesley, Wokingham, UK, 1987.

D. J. Bower, ‘Currently Available Software For Paratransit Applications’, Time Capsule, Vol.4, Nos.1,2, pp.12-17, Fall/Winter 1985.

F P. Brooks, ‘The mythical man-month, essays on software engineering’, Addison-Wesley, Reading, Massachusetts, 1975.

C. Brown, ‘Rural Transport Demonstration Project’, Seminar Paper circulated at ATCO Summer Meeting, 9 July 1982.

A. Browne, ‘Report on the Multi Trip CT System’, The Link Organisation, Internal Report, 1992.

I Buchanan and McKinnon, K. ‘An animated interactive modelling system for decision support’, European Journal of Operational Research, Vol.54, pp.306-317, 1991.

M. Bull, ‘The PICK Operating System’, Chapman and Hall, London, 1986.

K. Campbell, ‘Using computers to monitor and analyse dial-a-ride performance’, in J. C. Sutton with Gillingwater, D. (eds), ‘Computer Applications in Community Transport - Which Way Forward?’, Department of Transport Technology, University of Technology, Loughborough (TT8703), July 1987.

Carter-Goble Associates Inc., 'Transportation Management Information Systems: South Carolina Case Study', Carter-Goble Associates Inc., Columbia, SC, USA, 1982.

I. Catling, 'Autoguide: A New Generation of Traffic Control', Engineering Design Education and Training, pp 2-6, Autumn 1988.

P. P-S. Chen, 'The Entity-Relationship Model - Toward a Unified View of Data', ACM Transactions on Database Systems, Vol.1, No.1, pp.9-36, March 1976.

B. Chown, 'Computer-assisted Management of Special Transit Services for the Disabled', Fifth International Conference on Mobility and Transport for Elderly and Handicapped Persons, Stockholm, 1989.

P. A. Clark, 'Action Research and Organizational Change', Harper and Row, London, 1972

E. F. Codd, 'A Relational Model of Data for Large Shared Data Banks', Communications of the ACM, Vol.13, No 6, June 1970

S. S. Cohen, 'Operational Research', Edward Arnold, London, 1985.

R. Cook and Brandon, J , 'The PICK Operating System. Part 1· Information Management', Byte, p.177, October 1984.

R Cook and Brandon, J., 'The PICK Operating System. Part 2. System Control', Byte, p.132, November 1984.

S. Cooke and Slack, N., 'Making Management Decisions', Prentice Hall International, Hemel Hempstead, 2nd Ed., 1991.

J. C. Cooper, 'The Use of Straight Line Distances in Solutions to the Vehicle Scheduling Problem', Journal of the Operational Research Society, Vol.34, No.5, pp 419-424, 1983.

Council for Science and Society, 'ACCESS FOR ALL? Technology and Urban Movement - Report of a Working Party', The Council for Science and Society, London, 1986.

R M Curtice and Jones, P. E , 'Logical Data Base Design', Van Nostrand Reinhold, New York, 1982

M. R. Cutler and Harman, L. J., 'The Application of Commercially Available Software to Paratransit Management', in 'Third International Conference on Mobility and Transport of Elderly and Handicapped Persons - Conference Proceedings', US Department of Transportation, Washington, October 1984.

C. Davidson, 'Measuring UP', Computer Weekly, p.28, November 7 1991.

H. E. H. Davies, 'The East Sussex Transport Broker: A Preliminary Report', Transport and Road Research Laboratory Report RR88, Crowthorne, 1987.

J. Dayhoff, 'Neural Network Architectures: An Introduction', Van Nostrand Reinhold, New York, 1990

E. Downs, Clare, P. and Coe, I, 'Structured Systems Analysis and Design Method: Application and Context', Prentice Hall International, Hemel Hempstead, Hertfordshire, 1988.

P. Drucker, 'What Business Can Learn from Nonprofits', Harvard Business Review, July-August 1989.

Dynatrend, 'Commercial Software Applications for Paratransit', US Department of Transportation, Washington DC, USA, 1984.

R. J. Elliott and Lesk, M. E., 'Route Finding in Street maps by Computers and People', Proceedings of the American Association for Artificial Intelligence, pp 258-261, 1982.

P. Evans, 'Management Information Systems', Community Transport Services Research Unit Sixth Annual Conference, Nottingham, June 1989.

S. R. Evans and Norback, J P., 'The Impact of a Decision-Support System for Vehicle Routeing in a Foodservice Supply Situation', Journal of the Operational Research Society, Vol.36, No.6, pp.467-472, 1985.

N. G. Fielding and J L Fielding, 'Linking Data', Sage University Paper series on Qualitative Research Methods, Volume 4, Sage, Beverly Hills, California, 1986.

Fletcher Computer Services, 'Dial-a-Ride', Fletcher Computer Services, Birmingham, 1984.

P. Ford, 'Derby Winner', Community Transport Magazine, Vol. 7, No 6, p.38, 1988

P. Ford and Forkin, J., 'Utilising data base management systems in CT group hire', in J. C. Sutton with Gillingwater, D. (eds), 'Computer Applications in Community Transport - Which Way Forward?', Department of Transport Technology, University of Technology, Loughborough, Report No. TT8703, July 1987.

R. Forsyth, 'Expert Systems: Principles and Case Studies', Chapman and Hall, London, 1984.

R. Forsyth and Rada, R., 'MACHINE LEARNING: Applications in expert systems and information retrieval', Chapman and Hall, London, 1984.

W. O. Galitz, 'Handbook of Screen Design Format', QED Information Sciences Inc , Wellesley, MA, USA, 1985

T. Geehan, 'Applications of Computers to Special Transit Services. A Review', Transportation Development Centre and BC Transit, Montreal, Canada, 1985.

R. V Giangrande and Harman, L. J., 'Microcomputer Applications in the Management of Paratransit Operations', in N J. Ashford, Bell, W. G. and Rich, T. A., 'Mobility and Transport for Elderly and Handicapped Persons', Gordon and Breach, 1982.

R. V. Giangrande, 'Generic Small Transit Data Management Software Program (TRANSIT) Demonstration Version 1.0', Department of Transportation/Transportation Systems Center, Cambridge, Massachusetts, USA, March 1985.

T. Gilb and Weinberg, G. M., 'Humanized Input: Techniques for Reliable Keyed Input', QED Information Sciences Inc., Wellesley, MA, USA, 1984.

D. Gillingwater, 'Resourcing Non-Conventional and Accessible Public Transport in the U.K ', Department of Transport Technology, University of Technology, Loughborough (TT8803), 1988.

D. Gillingwater and Bennett, S. C., 'Bedarfsgesteuerter Verkehr am Beispiel MULTI TRIP', Intensivkurs für rechnergestutzter Systeme fur den offentlichen Nahrverkehr, Hanover, Germany, 1989.

D Gillingwater and Sutton, J., 'Information technology applications in voluntary sector transport SP1: Objectives and Programme of Work', Department of Transport Technology, University of Technology, Loughborough (TT8501), 1985.

D. Gillingwater and Sutton, J. C., 'Information technology applications in voluntary sector transport. (WP2 Project Implementation)', Department of Transport Technology, University of Technology, Loughborough (TT8601), 1986.

D. Gillingwater and Sutton, J. C., 'Information technology applications in voluntary sector transport: WP4 Final Report', Department of Transport Technology, University of Technology, Loughborough (TT8801), 1988.

B. L. Golden and Wasil, E. A., 'Computerized vehicle Routing in the Soft Drink Industry', Operations Research, Vol.35, No.1, pp.6-17, 1987.

G. A. Gorry and Scott-Morton, M. S., 'A framework for management information systems', Sloan Management Review, Vol.13, No 1, pp 56-70, 1971.

T. J. Grant, 'Lessons for O.R. from A.I. - A Scheduling Case Study', Journal of the Operational Research Society, Vol.37, No.1, pp.41-57, 1986.

J. R Groff and Weinberg, P. N., 'Using SQL', Osborne McGraw-Hill, Berkeley, California, USA, 1990.

R Guindon, 'Cognitive Science and its Applications for Human-Computer Interaction', Lawrence Erlbaum Associates, Hillsdale, New Jersey, U S A , 1988.

Hackney C&T Systems, 'Proposal 1985', Hackney Community and Transport Systems, Hackney, 1985.

N. Hamer, 'Implementation of Computer Systems for Specialized Transit: Three Case Studies', GIRO Inc., Montreal, Canada, 1986.

N. Hamer and Rousseau, J-M , 'Microcomputer Applications in Smaller Paratransit Agencies', in W. G. Bell and Ashford, N. J. (eds), 'Third International Conference on Mobility and Transport of Elderly and Handicapped Persons', US Department of Transportation, Washington DC, USA, 1984.

P Harmon and King, D , 'Expert Systems: Artificial Intelligence in Business', John Wiley and Sons, 1985.

J Hopfield and Tank, D. 'Computing with Neural Circuits: A Model', Science, Vol 233, pp.625-633, August 1986.

D. R Howe, 'Data Analysis for Data Base Design', 2nd Ed., Edward Arnold, London, 1989.

J-J. Jaw, Odoni, A. R., Psaraftis, H. N. and Wilson, N. H. M , 'A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows', Transportation Research B, Vol.20B, No.3, p.243-257, 1986.

D. J. Jeffrey, Russam, K. and Roberston, D. I., 'Electronic route guidance by AUTOGUIDE: the research background', Traffic Engineering and Control, Vol.28, No.10, pp.525-529, 1987.

P. N. Johnson-Laird, 'The Computer and the Mind', Fontana, London, 1988.

E. Jones, 'What Can Computers Do for Us?', Community Transport Magazine, Vol.7, No 3, p.39, 1988.

E. Jones, 'Help is at Hand', Community Transport Magazine, Vol.8, No.2, pp.25-26, 1989

S. Kagaya, Kikuchi, S. and Donnelly, R. A., 'Use of Fuzzy Theory Technique for Grouping of Trips in Vehicle Routing and Scheduling Problem', Paper accepted for publication in European Journal of Operational Research, December 1991.

S. Kikuchi, 'Alternative routing and scheduling schemes for specialized transportation vehicles', Specialized Transportation Planning and Practice, Vol 2, pp 303-323, 1987.

S. Kikuchi and Donnelly, R. A., 'Scheduling demand-responsive transportation vehicles using fuzzy-set theory', Journal of Transportation Engineering, Vol 118, No 3, pp.391-409, 1992.

A. W. J Kolen, Rinnooy Kan, A H. G. and Trienekens, H. W. J. M , 'Vehicle Routing with Time Windows', Operations Research, Vol.35, No.2, pp 266-273, 1987.

R. P. Knott, 'A Knowledge Based Vehicle Scheduler for Emergency Relief Management', Institution of Electrical Engineers Computing and Control Division Colloquium on ''Applications of Expert Systems in Road Transportation'', January 1988.

R I Levine, Drang, D. E. and Edelson, B., 'A Comprehensive Guide to AI and Expert Systems', McGraw-Hill, New York, 1986

J. D. C. Little, Murty, K. G., Sweeney, D. W. and Karel, C, 'An algorithm for the traveling salesman problem', Operations Research, Vol.11, No.6, pp.972-989, 1963.

LOLA (London On-Line Local Authorities), 'Haringey Community Transport - Proposal for Introducing a Micro-Computer System', LOLA, 1985.

T. Lucey, 'Management Information Systems', D P Publications, London, 6th Ed , 1991.

B. B. Mandelbrot, 'The Fractal Geometry of Nature', W. H. Freeman, 1983.

T. Mannion, Undated correspondence with Dr John Sutton on Runbrok 84 options.

N. Marsh and Jarrett, M. (ed), 'Computers in Voluntary Groups - Five Case Studies', National Council for Voluntary Organisations, London, 1985.

B. P. Mathews and Waters, C. D J., 'Computerized Routeing for Community Nurses - A Pilot Study', Journal of the Operational Research Society, Vol.37, No 7, pp.677-683, 1986.

R. E Mayer, 'Thinking, Problem Solving, Cognition', W. H. Freeman, New York, USA, 1983.

G A. Miller, 'The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Information Processing', Psychological Review, Vol.63, No.2, pp.81-96, March 1956

T. Meadows, 'The Computerisation of Dial a Ride', National Advisory Unit for Community Transport (NAU 141), Manchester, 1987.

J. G. Moser, 'Integration of artificial intelligence and simulation in a comprehensive decision support system', Simulation, Vol.47, No.6, pp.223-229, 1986.

National Advisory Unit for Community Transport, 'Study of Booking Systems - The Impact of the Booking System of Six London Dial-a-Rides on Trip Requests, Refusals and Cancellations', National Advisory Unit for Community Transport, London, 1985

A. Newell and Simon, H A , 'Human Problem Solving', Prentice-Hall, Englewood Cliffs, N.J., 1972.

NextBase, 'Autoroute User Manual', NextBase, Ashford, Middlesex, 1987

S. D. Nutley, '"Unconventional modes" of transport in the Unıted Kingdom - a review of types and the policy context', Transporation Research, Vol.22A, No.5, pp 329-344, September 1988.

R. E. Paaswell, McKnight, C. E. and Depa, J. V., 'An Introduction to Utilization of Microcomputers by Specialized Transportatıon Providers', in W. G. Bell and Ashford, N. J. (eds), 'Third International Conference on Mobılity and Transport of Elderly and Handicapped Persons', US Department of Transportation, Washington DC, USA, 1984.

T. Pettitt, 'Rural Transport Focus - Country Cars in Dyfed', AtcoNews, No.43, pp.13-14, October-November 1987.

R I. Phelps, 'Artificıal Intelligence - An Overview of Simılarıties with O R.', Journal of the Operatıonal Research Society, Vol.37, No.1, pp.13-20, 1986.

PICK Systems Inc., 'The PICK System R83 User Reference Manual, Rev. 4', PICK Systems, Irvine, Calıfornia, USA, 1988.

L K Platzman and Bartholdı, J. J., 'Spacefillıng curves and the planar traveling salesman problem', Georgıa Institute of Technology, School of Industrial Engineering, Atlanta, Georgia, February 1983.

S Potter and Hagyard, T., 'Dıal-a-Ride Despatchıng Guıde', Federatıon of London Dıal-a-Rıdes, London, 1986

E. Rich, 'Artıfıcıal Intelligence', McGraw-Hill Internatıonal Editıons, 1983.

F Samısh, 'Languages to open several doors', Multı-User Computıng, pp.18-24, January 1988

P. S. Sell, 'Expert Systems - A Practıcal Introductıon', Macmıllan, London, 1985

S Shlaer and mellor, S. J., 'Object-Orıented Systems Analysis: Modeling the World in Data', Yourdon Press, Englewood Cliffs, New Jersey, 1988.

H. Sımon, 'The New Science of Management Decısion', Harper and Row, New York, 1960.

A. U. Simpson, 'Some Recent Developments in Computer Technology for Transportation of Elderly and Handicapped Persons', in N. J. Ashford, Bell, W. G. and Rich, T. A., 'Mobility and Transport for Elderly and Handicapped Persons', Transportation Studies Volume 2, Gordon and Breach, 1982.

S. Smythe, 'Measuring Productivity in London Dial-a-Ride Services', Community Transport Magazine, Vol.7, No.1, pp.30-31, 1988.

S. Smythe, 'Counting Trips? - Practical Evaluation in the CT Sector', Community Transport Services Research Unit Eighth Annual Conference 'Evaluating Community Transport: Policy, Performance and Practice', Nottingham, June 1989.

M. M Solomon, 'Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints', Operations Research, Vol.35, No 2, pp 254-265, 1987.

G Spooner, 'What to Ask, Where to Go', Commercial Motor, 29th September 1988.

R H. Sprague and Carlson, E D., 'Building Effective Decision Support Systems', Prentice-Hall Inc , Englewood Cliffs, New Jersey, USA, 1982.

R. H. Sprague and Watson, H J. (eds.), 'Decision Support Systems Putting Theory into Practice', Prentice-Hall International, Englewood Cliffs, New Jersey, USA, 2nd Ed , 1989.

P. J. Stacey, 'Practical Vehicle Routeing Using Computer Programs', Journal of the Operational Research Society, Vol.34, No 10, pp.975-981, 1983.

J-A. Stone and Geehan, T , 'Computerized Scheduling/Brokerage Demonstration Project in Vancouver', British Columbia Transit and TransVision Consultants, Vancouver, BC, Canada, 1986.

M. Sugie, Menzilcioglu, O. and Kung, H T., 'CARGuide - on-board computer for automobile route guidance', AFIPS, Vol.53, pp 695-706, 1984.

J. C. Sutton, 'Computer Assisted Bus Systems', Trent Papers in Planning, No 18, Trent Polytechnic, Nottingham, 1980.

J C Sutton, 'IT in CT: National Computer Survey. Briefing Paper 1: Initial Results', IT in CT Project, Department of Transport Technology, Loughborough University of Technology, Loughborough, 1986

J. C. Sutton, 'A review of trends in computer applications in the CT sector', in J. C. Sutton with Gillingwater, D. (eds), 'Computer Applications in Community Transport - Which Way Forward?', Department of Transport Technology, University of Technology, Loughborough (TT8703), July 1987.

J. C. Sutton, 'Transport innovation and passenger needs - changing perspectives on the role of Dial-a-Ride systems', Transport Reviews, Vol.7, No.2, pp.167-182, 1987.

J. C. Sutton, 'Community transport services in the United Kingdom: history, organization and recent developments', Specialized Transportation Planning and Practice, Vol.2, pp.325-350, 1987.

J. C. Sutton, 'Transport Coordination and Social Policies', Avebury Press, Aldershot, 1988.

J. C. Sutton, 'Travel choice and the mobility handicapped: Some preliminary investigations utilising a discrete choice model', Fifth International Conference on Mobility and Transport for Elderly and Handicapped Persons, Stockholm, 1989

J. C. Sutton, 'Performance Evaluation: Setting the Scene', Community Transport Services Research Unit Eighth Annual Conference 'Evaluating Community Transport. Policy, Performance and Practice', Nottingham, June 1989.

J. C. Sutton with Gillingwater, D. (eds), 'Computer Applications in Community Transport - Which Way Forward?', Department of Transport Technology, University of Technology, Loughborough (TT8703), July 1987.

J C. Sutton, Gillingwater, D. and Bennett, S C., 'Booting Up. A Novice's Guide to IT in CT, IT in CT Project, University of Technology, Loughborough, 1987.

J. C. Sutton, Gillingwater, D., Bennett, S. and Wong, S., 'Information Technology Applications in Dial-a-Ride. The Haringey Dial-a-Ride Scheme', Department of Transport Technology, University of Technology, Loughborough (TT8704), August 1987.

D W Sweeney, 'The Exploration of a New Algorithm for Solving the Traveling Salesman Problem', M.S. Thesis, M.I T., 1963.

E J. Taaffe and Gauthier, H. L., 'Geography of Transportation', Prentice-Hall, Englewood Cliffs, N J., U.S.A., 1973.

M. Taylor, 'PICK for Users', Blackwell Scientific Publications, Oxford, UK, 1985.

M. A. P. Taylor, 'Knowledge-based systems for transport network analysis: a fifth generation perspective on transport network problems', Transportation Research A, Vol.24A, No.1, PP.3-14, 1990.

D. Teodorovic and Kikuchi, S., 'Application of fuzzy sets theory to the saving based vehicle routing algorithm', Civil Engineering Systems, Vol.8, pp 87-93, 1991.

Time Capsule, 'Small Transit Data Management Software (SST3)', Time Capsule, Vol.6, Nos.3,4, p.13, Spring/Summer 1988.

Transport Design and Development, 'TDD-ESCORT', Transport design and Development, Lewes, East Sussex, (Undated brochure).

Transport Design and Development, 'ESCORT File Structure', 1985.

E. Turban, 'Decision Support and Expert Systems: management support systems', Macmillan, New York, USA, 2nd Ed., 1990.

Urban Mass Transportation Administration, 'Microcomputers in Transportation: Software and Source Book', US Department of Transportation, Washington, DC, USA, 1984.

T. Van Vuren and Jansen, G. R. M., 'Recent Developments in Path Finding Algorithms A Review', Transportation Planning and Technology, Vol.12, pp 57-71, 1988.

R. Veryard, 'Pragmatic Data Analysis', Blackwell, Oxford, 1984

VIA Systems Inc., 'VIA Duct User Manual', 1991.

Y. Wærn, 'Cognitive Aspects of Computer Supported Tasks', John Wiley, Chichester, 1989.

A. Warrington, 'The Co-ordination of Community Transport Resources: A Case Study of "Shared Transport Service", Birmingham', Department of Transport Technology, University of Technology, Loughborough (TT8603), 1986.

A. J. Warrington, Bryman, A. and Gillingwater, D., 'The Co-ordination of Community Transport Resources: A Case Study of the Minibus Pooling Scheme Administered by Tower Hamlets Community Transport', Department of Transport Technology, University of Technology, Loughborough (TT8701), March 1987

P. D. Wasserman, 'Neural Computing: Theory and Practice', Van Nostrand Reinhold, New York, 1989.

C. D J. Waters, 'Interactive Vehicle Routeing', Journal of the Operational Research Society, Vol 35, No.9, pp.821-826, 1984.

C. D. J. Waters, 'Expert Systems for Vehicle Scheduling', Journal of the Operational Research Society, Vol.41, No.6, pp.505-515, 1990.

P. F. Watts, 'Co-ordinated Public Transport in East Sussex: County Rider Services 823 and 825', Transport and Road Research Laboratory Report RR7, Crowthorne, 1985

M. D. Wheatley, 'Expert Systems in Transport. Part 1: An introduction to expert systems', Working Paper 178, Institute for Transport Studies, University of Leeds, Leeds, 1984.

W. A. Wickelgren, 'How to Solve Problems', W. H. Freeman, San Francisco, U.S A , 1974.

J. Williamson, 'Scheduling by Computer', Community Transport Magazine, Vol.6, No 4, p.15, 1987.

W. L. Winston, 'Operations Research: Applications and Algorithms', Duxbury Press, Boston, U S A., 1987.

J. Wolfenden (Baron Wolfenden), 'The Future of Voluntary Organisations - Report of the Wolfenden Committee', Croom Helm, London, 1978.

Wootton Jeffreys and Partners, 'Community Transport - Computer Assisted Fleet Administration System', Wootton Jeffreys, 1980.

T Wren (ed), 'Computer Scheduling of Public Transport', North Holland Publishing, Amesterdam, Netherlands, 1981.

E. Wyatt and Smerk, G., 'Microcomputers in Transit: A Software Handbook', US Department of Transportation, Washington, DC, USA, 1984

M. Yazdani, 'Knowledge Engineering in PROLOG', in R. Forsyth, 'Expert Systems: Principles and Case Studies', Chapman and Hall, London, 1984.

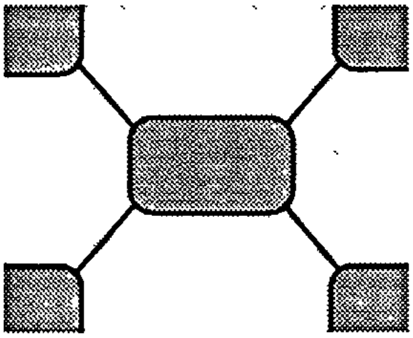
S. Zuboff, 'In the age of the smart machine, the future of work and power', Heinemann Professional, Oxford, 1988.

Appendix A

MULTI TRIP System Design Summary

June 1992

MULTI TRIP



SYSTEM DESIGN SUMMARY

Copyright MULTI TRIP Software Ltd., June 1992

MULTI TRIP SYSTEM DESIGN SUMMARY

Copyright MULTI TRIP Software Ltd , June 1992

A. Contents

B. Introduction

1. Overview

This document is copyright MULTI TRIP Software Ltd, June 1992.

This document is written to provide an overview of MULTI TRIP, a modular database package which provides support for the operational and management functions of a community transport operator. MULTI TRIP was developed by a research project in the Department of Transport Technology, University of Technology, Loughborough, UK. The research was funded by the Science and Engineering Research Council between September 1986 and March 1989 (ref.GR/E/01317), "The application of a modular IT system to community transport management and transport brokerage projects".

Community transport in the sense in which it is applied in the UK means a sector of transport operators, with the following characteristics:

most are voluntary organisations or charities,
most are publicly funded, and
they provide a range of transport services for voluntary and community organisations and for people with disabilities.

These services include the following

group hire - minibus hire for groups; *
vehicle brokerage - minibus hire on behalf of vehicle owners; *
dial-a-ride - many-to-many transport for people with disabilities; *
dial-a-bus - many-to-one transport for people with disabilities,
furniture - transport of furniture or hire of vans;
social car - volunteer drivers using their own vehicles to transport people with disabilities *

Some operators only offer one such service, while others provide more than one.

MULTI TRIP is intended eventually to cater for all these services, and currently caters for those marked with an asterisk, to some degree.

2. Purpose

The purpose of this document is to provide for GIS/Trans Ltd, and for the Link Organisation, a summary and reference document for MULTI TRIP in relation to the following aspects of the design:

Software environment,
Hardware options;
Modular structure,
Data structures;
Program flow of control

3. Other Documents

There are a number of other documents available which provide documentation in relation to MULTI TRIP.

a) SERC Proposal
The original proposal to the SERC for the research project.

b) "Designing a modular software package dedicated to CT group hire, dial-a-ride and transport brokerage", in "Computer Applications in Community Transport - Which Way Forward", TT8703, Loughborough University, 1987

c) Final Report
"Application of a modular IT system to community transport management and transport brokerage projects".

d) MULTI TRIP Manual
User Manual for MULTI TRIP.

4. Abbreviations

The following abbreviations are used in this document:

CT - Community transport
GH - Group Hire
DAR - Dial-a-Ride
DAB - Dial-a-Bus
SC - Social Car

C. Software

MULTI TRIP has been developed using and to take advantage of the facilities of the PICK operating system.

1. PICK

MULTI TRIP has been developed and written for the PICK operating system PICK is available in a number of varieties: these are produced by licensees who are predominantly hardware companies who have been licensed by PICK Systems to sell an operating system derived from PICK on their machines

The PICK OS provides a multi-user, virtual memory, database management environment and 3rd and 4th generation programming languages

a) DATA/Basic
A version of Dartmouth BASIC optimised to the PICK DBMS file structure. Compiled to a virtual machine code which is interpreted at run-time by the OS

b) ACCESS
An enquiry language, which provides a natural language interface to data held in the PICK DBMS. A precursor to SQL, and not compatible with it

c) PROC
A job control language, with interactive capabilities to, for example, prompt for parameters which can be built into an ACCESS statement or passed in a buffer to a DATA/Basic program.

2 System Builder

The System Builder applications generator has been used to develop data entry, enquiry and transaction screens and processes, and reports and printed documents. Version 4 2 of System Builder was used to develop the package. Programs developed using System Builder and used in the software package are in one of three modes

a) Parameter Driven
System Builder provides a run-time interpreter which translates parameters which define screens, validations and update processes into 'drivers' which are interpreted

Much functionality within the software is provided by calling external subroutines written in DATA/Basic from within parameter driven or generated programs.

b) Generated Code
System Builder can generate DATA/Basic program code from the parameter files for data entry, enquiry, transaction update and document print processes. These programs can then be compiled and run, in this case from within the menu framework provided by System Builder, in the same way as any other DATA/Basic program.

c) Amended Generated Programs
For a small number of programs, the System Builder generated code has been amended to provide functionality which cannot be provided by System Builder or by the calling of external subroutines.

D. Hardware

MULTI TRIP will run on a wide range of hardware platforms which support the PICK OS. Currently it is in use on the following three platforms.

1. PC

PICK Systems provides PICK for the IBM PC, both for XT and for AT class (including 386) machines, and for the PS/2.

PICK is installed in a separate partition on the hard disk and can reside with DOS, although they cannot be used concurrently.

2. C-Itoh

68xxx processor machines running a licensed version of PICK.

3. ADDS Mentor

ADDS is a subsidiary of NCR, and the 6000 series is based on the NCR Tower, running PICK instead of UNIX. Mentor is ADDS name for their licensed version of PICK.

4 Terminals

PICK supports asynchronous full-duplex terminals connected over RS232 connections. PC's with appropriate terminal emulation software can be used as terminals. PICK supports terminals through configurable drivers, and a utility to maintain these drivers is provided, as well as a selection of drivers for common terminals. System Builder holds a separate set of terminal definitions, which are also configurable.

Terminal screen and character attributes and cursor movements can be coded into DATA/Basic programs and PROCs in a form which is translated dynamically by the drivers into the control sequence for the terminal in use on a particular serial port.

5. Printers

Generic PICK does not support printer drivers, but C-Itoh PICK supports drivers which allow the control of fonts and print features at print time.

6. Floppy Disk Drives

Floppy disks are used for back-up of accounts and files, and are treated by PICK as serial media, blocks of data being written to successive sectors.

7. Tape Drives

While larger PICK machines support 1/2" tape, most PICK machines support 1/4" 60Mb DC600 tape cartridges Mentor supports 150Mb cartridges.

E. System Overview

MULTI TRIP provides operational and management functions for operators of community transport (CT) services in the UK. A distinction is made in the operational functions between those which are common to all modes of CT service and those which are specific to particular modes of service delivery. The way in which these functions are separated out is described in the next section on the Modules of MULTI TRIP.

MULTI TRIP is a menu-driven package, using the menu structure provided by System Builder Most data entry, data enquiry, printing and reporting functions can be accessed using the menus

In addition a number of functions concerned with the making and amending of bookings can be accessed from within the bookings diary, which is navigated using the numeric keypad and single letter mnemonics.

F Modules

MULTI TRIP consists of the following applications modules.

1. Common Module

The Common Module provides those functions which are common to all modes of CT service

a) Mailing List
Maintenance of names and addresses of individuals who may be drivers, member group contacts or simply recipients of mailed information.

b) Own Use Bookings
Maintenance of bookings for vehicles by vehicle owners, either the operator or owners of vehicles in brokerage, for own use, such as maintenance.

c) Log Sheet Entry
Entry of vehicle log sheets after trips, which form basis of invoicing for group hire and of general reporting procedures on vehicle use

d) Drivers
Maintenance of records of staff and volunteer drivers for all services.

e) Vehicles
Maintenance of records of vehicles for all services. Maintenance of vehicle availability charts and diary.

f) Code Files
Maintenance of code files used throughout the system.

g) Data Protection Act
Print-outs of personal data held about identifiable individuals and as required by the subject access provisions of the Data Protection Act 1984.

h) Set-Up
Maintenance of site specific data, such as name and address of operator and a wide range of operating parameters

i) File Management
Disk housekeeping options.

j) Ad Hoc Reporting
Reporting using System Builder Development version report generator. (Not available if user only has Runtime version)

2. Group Hire/Brokerage

The Group Hire Module provides the functions required for the operation of a group hire service, including vehicle brokerage.

a) Member Groups
Maintenance of details of member groups, including vehicles owned in brokerage, drivers and contacts

b) Group Hire Bookings
Maintenance of individual bookings for member groups and of regular bookings which are used as templates to generate individual bookings automatically

c) Group Hire Drivers
As 1 (d) above with reports specific to group hire.

d) Group Hire Vehicles
As 1 (e) above with reports specific to group hire and charge rates for use of vehicles in group hire service.

e) Group Passenger Lists
Maintenance of lists of passengers for group hire bookings.

f) Vehicle Brokerage
Maintenance of individual bookings for vehicles in brokerage, with maintenance of accounts for vehicle owners.

g) Vehicle Operators
Maintenance of records of other, commercial vehicle operators.

3. Dial-a-Ride

The Dial-a-Ride Module provides the functions required for the operation of a dial-a-ride service.

a) Passenger Service Users
Maintenance of users of passenger services.

b) Dial-a-Ride Bookings
Maintenance of users' bookings for dial-a-ride.

c) Dial-a-Ride Shifts
Maintenance of daily dial-a-ride vehicle shifts and of regular vehicle shifts which are used as templates to generate daily shifts automatically.

d) Dial-a-Ride Charges
Maintenance of charge rates and concessions for dial-a-ride service users.

e) Passenger Assistants and Escorts
Maintenance of staff and volunteer assistants and escorts for passenger service users.

f) Dial-a-Ride Codes
Dial-a-Ride specific code file maintenance.

4. Social Car Scheme

The Social Car Scheme Module is the most recent and least developed module within the package. It provides the functions required for the operation of a social car service.

a) Passenger Service Users
Maintenance of users of passenger services

b) Social Car Bookings
Maintenance of users' bookings for social car.

c) Social Car Shifts
Maintenance of daily social car vehicle shifts and of regular vehicle shifts which are used as templates to generate daily shifts automatically.

d) Social Car Charges
Maintenance of charge rates and concessions for social car service users.

e) Passenger Assistants and Escorts
Maintenance of staff and volunteer assistants and escorts for passenger service users.

f) Social Car Codes
Social Car specific code file maintenance.

5. Accounts

The Accounts Module provides basic accounting functions linked to the other modules

a) General Ledger
Nominal/General ledger and Cash Book transactions.

b) Member Group Accounts
Sales ledger type facilities for invoicing group hire users for services, and maintaining accounts.

c) Vehicle Brokerage Accounts
Purchase ledger and sales ledger type facilities for recording invoices from brokerage owners and issuing invoices to them for commission or services provided.

d) Passenger Service Accounts
Sales ledger type facilities for invoicing account-holders on behalf of passenger service users. Account-holder may be other than service user, e.g. social services department.

e) Social Car Driver Accounts
Purchase ledger type facilities for payments to volunteer drivers in social car scheme.

6. Report-Wise

Report-Wise provides ad-hoc reporting capabilities which are not dependent on the availability of the System Builder Development version report generator. It produces reports which are run using the PICK ACCESS enquiry language It provides a form-filling front-end to ACCESS, and displays the generated enquiry before running it, as a training aid.

Report-Wise was written by Wise Software Ltd., and the right to include it with copies of MULTI TRIP was assigned to the MULTI TRIP Trust by Wise Software Ltd.

G. Data Structures

1. Files

Within PICK, most files contain two levels, the dictionary level and the data level. Within each PICK Account, the Master Dictionary (MD) contains a pointer to the dictionary level of each file in that account, and potentially to files in other accounts (Q-pointers), entries which represent TCL (operating system level commands such as CREATE-FILE), entries which represent ACCESS commands (such as LIST and SORT) and the ACCESS connectives (TOTAL etc.), and CATALOG entries for compiled programs enabling them to be run as operating system commands or to be called from other programs if they are subroutines.

It is possible to create a file which only has a dictionary level, for example as a POINTER-FILE which contains pointers to lists.

It is possible to create multiple data files which use the same dictionary level.

a) Dictionaries

Program files have a dictionary level and a data level. The data level contains the source code, and can be cleared or deleted once the programs have been compiled. The dictionary level contains pointers to the location of the object code Each pointer also contains the date and time of compilation, and has the same name as the source item In most versions of PICK, attribute 1 of the entry in the MD which defines the file must be 'DC' rather than 'D' if the dictionary is to contain such pointers

Data files have a dictionary level and a data level. The dictionary level contains entries which define the attributes (fields) of the items (records) in the data level. The structure of the dictionary items is as follows·

Attribute	Values	Function
1	A or S	Identifies an attribute dictionary item is for Synonym
2	0 or +ve integer	Attribute number in data item 0 is item-id (key)
3	Name	Used as column heading in reports
4		Controlling and dependent value indicator
5		
6		
7		Conversion code - for example D converts internal date values (no of days since 31st Dec. 1967) to external format.
8		Correlative code - for example arithmetic performed on stored value or join to another file.
9	L,R,T,U	Justification - Left, Right, Text, Unconditional
10	+ve Integer	Width of field when displayed

These dictionary items are primarily for use by the ACCESS processor to convert and format data correctly for output. In contrast to SQL, the ACCESS language does not support dynamic JOINs. To join two files, synonym dictionary items which use correlatives must be created

System Builder extends the PICK dictionaries The System Builder field definition program creates PICK dictionary items and its own items. The name of each System Builder dictionary is the same as the PICK one, but prefixed by a dot, '.', and the first attribute is a 'Z'. The System Builder dictionary items contain up to 16 attributes instead of the 10 of the standard dictionary items System Builder also uses the dictionary of a file to store screen definitions for data entry and enquiry screens on that file; for these items, the first attribute is the word 'SCREEN' System Builder also uses the dictionary for print form definitions for print-outs based on that file; for these items, the first attribute is the word 'PRINT'.

System Builder also creates entries in the dictionary prefixed with the dollar character, '$', which perform special functions. These are.

$CROSS.REF
Cross reference information used to build the secondary index on the file.
$LAST.POS
The attribute number of the last attribute created using the System Builder field definition program.
$KEY
A copy of the dictionary item for the item-id or key
$DUMMY1 etc.
Dictionary items created to allow the creation of selections from the file based on equality between different attributes

b) Data

The data level of each file contains items (records) which are made up of variable length fields and which are themselves variable length In most versions of PICK, there is a 32k limit on the size of each item Each field is called an attribute and may be defined in the dictionary. Each attribute may contain more than one value, and each value may contain more than one sub-value. This 3-dimensional array structure can be referenced by the use of variable subscripts within programs. System Builder supports the use of multiple values, but not sub-values It is normal practice in PICK systems to implement repeating groups as multiple values rather than fully normalising data.

Attributes, values and sub-values are delimited by special characters·

ASCII value	Name	Function	Shows as
255	Segment marker	End of item in file	_
254	Attribute marker	Delimits attributes	^
253	Value marker	Delimits values	]
252	Sub-value marker	Delimits sub-values	\

c) Files and details

Q Type files are files in other PICK accounts which can be accessed from the MULTI TRIP account. D Type files are data files   DC Type files are either pointer files for holding lists, or program files. All the MULTI TRIP program files are in the account MTPROGS.

The following files are used in MULTI TRIP:

FILENAME	TYPE	DESCRIPTION
ACC	Q	POINTER TO ACC IN ACC
CTPROGS	Q	POINTER TO CTPROGS IN MTPROGS
DM	Q	POINTER TO DM IN SB4.2
DMAD	Q	POINTER TO DMAD IN SB4.2
DMCONTROL	Q	POINTER TO DMCONTROL IN SB4.2
DMDOC	Q	POINTER TO DMDOC IN SB4.2
DMGS	Q	POINTER TO DMGS IN SB4.2
DMHELP	Q	POINTER TO DMHELP IN SB4.2
DMOT	Q	POINTER TO DMOT IN SB4.2
DMSKELETON	Q	POINTER TO DMSKELETON IN SB4.2
DMTS	Q	POINTER TO DMTS IN SB4.2
DMTSCONTROL	Q	POINTER TO DMTSCONTROL IN SB4.2
DMTSHELP	Q	POINTER TO DMTSHELP IN SB4.2
DMTSMAIL	Q	POINTER TO DMTSMAIL IN SB4.2
DMTUTTEXT	Q	POINTER TO DMTUTTEXT IN SB4.2
DMUTIL	Q	POINTER TO DMUTIL IN SB4.2
DMVERBS	Q	POINTER TO DMVERBS IN SB4.2
DRPROGS	Q	POINTER TO DRPROGS IN MTPROGS
ERRMSG	Q	POINTER TO ERRMSG IN SYSPROG
MTPROCS	Q	POINTER TO MTPROCS IN MTPROGS
RW BP	Q	POINTER TO RW.BP IN MTPROGS
SBP	Q	POINTER TO BP IN SYSPROG
SCPROGS	Q	POINTER TO SCPROGS IN MTPROGS
SUBPROGS	Q	POINTER TO SUBPROGS IN MTPROGS
SYSPROG-BP	Q	POINTER TO BP IN SYSPROG
UPGRADE	Q	POINTER TO UPGRADE IN MTPROGS
WSLUFO	Q	POINTER TO WSLUFO IN MTPROGS
CTWORK	DC	POINTER FILE FOR LISTS
POINTER-FILE	DC	SYSTEM POINTER FILE
ACLEDG ARCH	D	GENERAL LEDGER ARCHIVE
ACLEDG TRANS	D	GENERAL LEDGER TRANS.
ACLEDG.TRANS ARCH	D	LEDGER TRANS.ARCHIVE
ACLEDGER	D	GENERAL/NOMINAL LEDGER
ALTEREDPROGS	D	AMENDED GENERATED PROGRAMS
ARCHIVES	D	ARCHIVE PARAMETERS
ARCHIVES.XREF	D	INDEX FOR ARCHIVE PARAMETERS
CTASSISTANTS	D	DRIVER'S ASSISTANTS
CTAVAIL.ASST	D	ASSISTANT AVAILABILITY
CTAVAIL.DRIV	D	DRIVER AVAILABILITY

CTAVAIL.MTH	D	AVAILABILITY BY MONTH
CTAVAIL.VEH	D	VEHICLE AVAILABILITY
CTBARNSLEY.AZ	D	A-TO-Z REFS FOR BARNSLEY
CTBATCH	D	SYSTEM BUILDER BATCH FILE
CTBATCHDATA	D	SYSTEM BUILDER BATCH DATA FILE
CTBOOKINGS	D	OWN USE BOOKINGS
CTCONTROL	D	SYSTEM BUILDER CONTROL FILE
CTDAMAGE	D	DAMAGE RECORDS
CTDIARY	D	VEHICLE BOOKING DIARY
CTDRIVER	D	DRIVERS (NOTE NOT PLURAL)
CTDRIVERS	D	SYSTEM BUILDER SCREEN DRIVERS
CTEXPENSES	D	DRIVER EXPENSES
CTFUEL	D	FUEL RECORDS
CTHELP	D	SYSTEM BUILDER HELP FOR MT
CTLANDMARKS	D	COMMON DESTINATIONS ADDRESSES
CTLISTS	D	LISTS OF VEHICLES ETC
CTLOGTRANS	D	LOGSHEET TRANSACTIONS
CTLONDON.AZ	D	LONDON A TO Z PAGES
CTLOOKUP.SFC	D	LOOKUP TABLE FOR MAP REFS
CTMAILING	D	MAILING LIST AND CONTACTS
CTOPERATORS	D	OTHER VEHICLE OPERATORS
CTREGULARS	D	OWN USE REGULAR BOOKINGS
CTREPORTDEFN	D	SYSTEM BUILDER REPORT DEFINITIONS
CTSYSTEM PARAMETERS	D	MULTI TRIP SYSTEM PARAMETERS
CTTEMPLATES	D	MT PROGRAM TEMPLATES
CTTESTDEFN	D	SYSTEM BUILDER TEST DEFINITIONS
CTTRANDEFN	D	SYSTEM BUILDER TRANSACTION DEF'NS
CTTRIPS	D	CT VEHICLE TRIPS
CTTUTTEXT	D	MULTI TRIP TUTORIAL TEXT
CTUPDATEDEFN	D	SYSTEM BUILDER UPDATE DEF'NS
CTUSERS	D	USERS OR PASSENGERS
CTVEHICLES	D	VEHICLES
CTVEHMAINT	D	VEHICLE MAINTENANCE RECORDS
CTVERBS	D	SYSTEM BUILDER COMMANDS
DOCUMENTS	D	SYSTEM BUILDER GENERATED DOCUMENTATION IN RUNOFF FORMAT
DRBOOKINGS	D	DIAL-A-RIDE BOOKINGS
DRCHARGES	D	DIAL-A-RIDE CHARGES
DRMAPREF	D	DIAL-A-RIDE GRID MAPREFS
DRSHIFT.REGULARS	D	DIAL-A-RIDE REGULAR SHIFTS
DRSHIFT.TRANS	D	DIAL-A-RIDE SHIFT RELATED TRANSACTIONS
DRSHIFTS	D	DIAL-A-RIDE SHIFTS
DRZONE.ZONE	D	ZONE TO ZONE CHARGE RATES
GENPROGS	D	PROGRAMS IDENTIFIED BY WISE SOFTWARE AS MODIFIED
GPACCOUNTS	D	GROUP MEMBER ACCOUNTS
GPACNT.ARCH	D	GROUP ACCOUNTS ARCHIVE
GPACNT.TRANS	D	GP ACCOUNT TRANSACTIONS
GPACNT.TRANS.ARCH	D	GROUP TRANS. ARCHIVE
GPBOOKINGS	D	GROUP HIRE BOOKINGS
GPCHARGES	D	GROUP HIRE CHARGE RATES
GPINVOICES	D	GROUP HIRE INVOICES
GPMEMBERS	D	MEMBER GROUPS
GPPASSLIST	D	PASSENGER LISTS
GPREGULARS	D	REGULAR GROUP BOOKINGS
PASSWORD	D	SYSTEM BUILDER PASSWORDS
PRINT.WORK.FILE	D	WORK FILE FOR PRINTED OUTPUT

PSACCOUNTS	D	PASSENGER ACCOUNTS
PSACCOUNTS.ARCH	D	PASSENGER ACCOUNTS ARCHIVE
PSACNT.TRANS	D	PASS. SERVICE A/C TRANS.
PSACNT.TRANS.ARCH	D	PASSENGER TRANS. ARCHIVE
PSINVOICES	D	PASS. SERVICE INVOICES
REPORTS	D	REPORT-WISE PARAMETERS
REPORTS.XREF	D	INDEX FOR REPORTS
SCACCOUNTS	D	SOCIAL CAR SCHEME A/CS
SCACNT.ARCH	D	SOCIAL DRIVER TRANSACTIONS ARCH.
SCACNT.TRANS	D	SOCIAL DRIVER TRANSACTIONS
SCACNT.TRANS.ARCH	D	SOC. DRIV. TRANS. ARCH.
SCBOOKINGS	D	SOCIAL CAR BOOKINGS
SCCHARGES	D	SOCIAL CAR CHARGE RATES
SCINVOICES	D	SOC. CAR DRIVER INVOICES
SCMAPREF	D	SOCIAL CAR GRID MAPREFS
SCSHIFT.REGULARS	D	SOCIAL CAR REGULAR BOOKINGS
SCSHIFT.TRANS	D	SOCIAL CAR SHIFT RELATED TRANSACTIONS
SCSHIFTS	D	SOCIAL CAR SHIFTS
SCTRIPS	D	DRIVER TRIPS WITH COSTS
SCZONE.ZONE	D	SOCIAL CAR ZONE CHARGES
VBACCOUNTS	D	VEHICLE BROKERAGE A/CS
VBACCOUNTS.ARCH	D	BROKERAGE ACCOUNTS ARCHIVE
VBACNT.TRANS	D	VEHICLE BROKERAGE TRANS
VBACNT.TRANS.ARCH	D	BROKERAGE TRANS. ARCHIVE
VBINVOICES	D	VEHICLE BROKERAGE INVOICES
VBTRIPS	D	VEHICLE BROKERAGE TRIPS
ACLEDG.TRANS.INDEX	D	INDEX TO ACLEDG.TRANS
ACLEDGER.INDEX	D	INDEX TO ACLEDGER
CTASSISTANTS.INDEX	D	INDEX TO CTASSISTANTS
CTAVAIL.DRIV.INDEX	D	INDEX TO CTAVAIL DRIV
CTBOOKINGS.INDEX	D	INDEX TO CTBOOKINGS
CTDAMAGE INDEX	D	INDEX TO CTDAMAGE
CTDRIVER.INDEX	D	INDEX TO DRIVERS FILE
CTEXPENSES INDEX	D	INDEX TO CTEXPENSES
CTFUEL.INDEX	D	INDEX TO CTFUEL
CTLANDMARKS.INDEX	D	INDEX TO CTLANDMARKS
CTMAILING.INDEX	D	INDEX TO MAILING LIST
CTREGULARS.INDEX	D	INDEX TO CTREGULARS
CTTRIPS.INDEX	D	INDEX TO CTTRIPS
CTUSERS.INDEX	D	INDEX TO CTUSERS
CTVEHICLES.INDEX	D	INDEX TO CTVEHICLES
DRBOOKINGS.INDEX	D	INDEX TO DRBOOKINGS
DRCHARGES.INDEX	D	INDEX TO DRCHARGES
DRSHIFTS.INDEX	D	INDEX TO DRSHIFTS
GPACNT.TRANS INDEX	D	INDEX TO GPACNT.TRANS
GPBOOKINGS INDEX	D	INDEX TO GPBOOKINGS
GPCHARGES INDEX	D	INDEX TO GPCHARGES
GPMEMBERS INDEX	D	INDEX TO MEMBER GROUPS
GPPASSLIST.INDEX	D	INDEX TO GPPASSLIST
GPREGULARS.INDEX	D	INDEX TO GPREGULARS
PSACNT.TRANS.INDEX	D	INDEX TO PSACNT.TRANS
SCACNT.TRANS INDEX	D	INDEX TO SCACNT.TRANS
SCBOOKINGS INDEX	D	INDEX TO SCBOOKINGS
SCCHARGES.INDEX	D	INDEX TO SCCHARGES
SCSHIFTS.INDEX	D	INDEX TO SCSHIFTS
VBACNT.TRANS.INDEX	D	INDEX TO VBACNT TRANS
ACDEPT.CODE	D	DEPARTMENT CODE FILE

ACLEDG CODE	D	GENERAL LEDGER CODES
CTABUSE.CODE	D	VEHICLE ABUSE CODE
CTACC.CODE	D	ACCIDENT/DAMAGE CODE
CTACNT.CODE	D	ACCOUNT TRANSACTION TYPES
CTADD.CODE	D	ADDRESS TYPE CODES
CTAID.CODE	D	MOBILITY AIDS CODE
CTATOZ.CODE	D	A TO Z PAGE DISPLACEMENTS
CTBROKER.CODE	D	BROKERAGE CODE
CTCANC.CODE	D	CANCELLATION CODE
CTCLAIM.CODE	D	ACCIDENT CLAIM CODE
CTDAY.CODE	D	DAYS OF WEEK
CTDIS.CODE	D	DISABILITY CODE
CTEND.CODE	D	DRIVER ENDORSEMENT CODE
CTETH.CODE	D	ETHNIC ORIGIN CODE
CTFAC.CODE	D	VEHICLE FACILITIES CODE
CTFARE.CODE	D	CONCESSIONARY FARE CODE
CTMAIL.CODE	D	CODES FOR MAILING LIST
CTMAINT.CODE	D	MAINTENANCE CODE
CTMOB.CODE	D	MOBILITY HANDICAP CODE
CTOCC.CODE	D	DRIVER OCCUPATION CODE
CTOWNER.CODE	D	VEHICLE OWNER CODE
CTOWNUSE.CODE	D	OWN USE CODE
CTPERM.CODE	D	DRIVER PERMIT CODE
CTREASON.CODE	D	REASON FOR LIMITATIONS
CTREF.CODE	D	REFUSAL CODE
CTREST CODE	D	VEHICLE RESTRICTION CODE
CTSERV.CODE	D	TYPE OF SERVICE CODE
CTSFC.CODE	D	SPACE FILLING CURVE
CTSOURCE.CODE	D	SOURCE OF INFORMATION CODE
CTSTATUS CODE	D	STATUS CODES PASSENGERS
CTSTOP.CODE	D	PICK-UP/DROP-OFF CODE
CTTEST.CODE	D	DRIVER TEST CODE
CTTRAIN.CODE	D	DRIVER TRAINING CODE
CTTRIP.CODE	D	TRIP TYPE CODE
CTVAT.CODE	D	VALUE ADDED TAX CODE
CTWC.CODE	D	WHEELCHAIR TRANSFER CODE
CTZONE CODE	D	ZONES AND MAP REFERENCES
GPACT.CODE	D	GROUP ACTIVITY CODE
GPFUND.CODE	D	GROUP FUNDING SOURCE CODE
GPORG.CODE	D	GROUP ORGANISATION TYPE CODE
GPPASS.CODE	D	GROUP HIRE PASSENGER CODE
GPPAYMT.CODE	D	GROUP PAYMENT METHOD CODE
GPPURP CODE	D	GROUP TRIP PURPOSE CODE
GPSTAFF.CODE	D	GROUP STAFF INITIALS CODE
GPSTAT.CODE	D	GROUP STATUS CODE
GPSUBS.CODE	D	GROUP SUBSIDY TYPE CODE

The following prefixes have been used

CT - common system
GP - group hire
DR - dial-a-ride
VB - vehicle brokerage
AC - Accounts
PS - passenger services
SC - social car

It is intended to use:

DB - dial-a-bus
PB - passenger brokerage
FG - furniture and goods

2. Variables

There is no data typing in PICK DATA/Basic.

System Builder uses a common block of variables, which contain data which is passed between programs which are chained, and between programs and called external subroutines.

Note: This information is proprietary information of System Builder Ltd.

The common block contains a mixture of arrays, flags and data values. The following are the names:

STEPS(250)	Array of interpreter steps in interpreted mode or screen fields in compiled mode.
STEPNO	Current STEP
CNT	Counter
NEWITEM	Flag for creating a new item
HEAD	Screen heading
REFRESH	Flag set when screen is to be refreshed
EXIT	Flag set when current loopis to be exitted
VAL	Input value
PORT	Port number on which program is running
SYSID	Two character mnemonic for system (CT for MULTI TRIP)
CONT	Control Account record - defines system - from file CTCONTROL in MULTI TRIP
PARMS(20)	Miscellaneous parameters - 11-20 are available for the developer and are used in MULTI TRIP. In particular, PARMS(20) contains the MULTI TRIP system parameters. PARMS(1) is used as an error flag, and must be set to 1:valid or 0:invalid by a called validation subroutine.
F.CONT	File variable on which control file (CTCONTROL) was opened
FILEVAR(19)	File variables to which data files are opened - 11-19 are free for use by the developer and are used in MULTI TRIP
KEY	Item-id of record currently being processed.
FLD(250)	Contains attributes (fields) of record currently being processed - maximum 250
F.MD	File variable to which Master Dictionary has been opened.
MENU.OPTION	Current menu option selected.
DRIVER.FNAME	Name of file containing driver item - in interpreted mode.
DRIVER.ID	Item-id of driver - in interpreted mode.
DRIV.VARS(15)	Other driver variables, such as screen.
OTHER(30)	Other miscellaneous parameters.

Further details of the common variables are contained in section 5 of the System Builder user guide and in the technical reference manual, which is held on the system and can be printed out.

Note that static array subscripts start at 1 not 0 Some versions of PICK will compile array subscripts of 0 and reference the previous variable in the symbol table!

In addition, certain values are usually set by EQUATE statements at the start of a program (equivalent to a 'C' #define) and are replaced by the compiler with the values to which they have been equated. For example:

```
EQUATE AM TO CHAR(254), VM TO CHAR(253), SVM TO CHAR(252)
EQUATE COMMA TO ',', ASTERISK TO '*'
EQUATE CLS TO @(-1), HOME TO @(-2)
```

Such equated variables will not appear in the symbol table and cannot be referenced in the debugger.

3. Diary

The structure of the data in the Diary (CTDIARY) is central to the vehicle scheduling capabilities of MULTI TRIP. The data is held on the file as an item containing attributes which are single data values, and attributes which consist of multiple values. Each item is keyed on the date in internal format (no. of days since 31 Dec 1967), concatenated to the vehicle no., with a slash '/' as a delimiter, e.g. 8740/0002. In multi-valued attributes, there is a value for every vehicle trip. These are:

```
001  BOOKING NO (Group Hire/Own Use Booking No. or
     Dial-a-Ride/Social Car Shift No.)
002  JOURNEY.NO (Vehicle Trip No.)
003  START TIME (In PICK internal format: seconds from midnight)
004  FINISH.TIME (ditto)
005  TYPE (G,D,S,O)
006  GROUP NO/USER NO (Group No. or DAR for Dial-a-Ride or SCS
     for Social Car Scheme)
```

In addition, Vehicle No and Date are held in attributes 7 and 8 The values are held in multi-values in time sequence It is not possible for trips to overlap in time.

The only subroutine which changes Diary items is DIARY in SUBPROGS, outlined in section H.7.f below The Diary item is read into a dimensioned array Checks on the Diary (using a relatively simple algorithm) are implementation dependent. In a pure relational database, the Diary structure would probably best be represented by a linked list. Multiple disk accesses would be required to read a single vehicle-day.

4. Dial-a-Ride and Social Car Schedules/Shifts

The structure of schedules for passenger trip based services provides a basis for scheduling passenger trips similar to that provided by the diary for scheduling vehicle trips. The data for a single shift is held in a single item, with multi-valued attributes which represent events during the course of the shift Thus any passenger trip is represented by at least two events, a pick-up and a drop-off Trips can overlap, to allow multiple occupancy, but a check is made on events occuring at the same time, and a flag set if the user specifies that pick-ups and drop-offs may occur together.

The shift record also forms the basis for producing drivers' worksheets and contains fare details as well as addresses, times, map references, zones, and the number of seated and wheelchair passengers joining or leaving the vehicle at any event

The structure of a shift record is:

Single valued attributes

001	DATE	Date of shift in PICK internal format
002	SHIFT.NO	Shift No.
003	SHIFT.LETTER	Shift Letter
004	START.TIME	Start time in PICK internal format
005	FINISH.TIME	Finish time in PICK internal format

Multi-valued - 1 value per stop event:

006	STOP.TIME	Event times
007	STOP.TYPE	Event types (PU/DO/BS/BF/DC)
008	BOOKING.NO	Booking Nos
009	STOP.ADDRESS	Addresses
010	MAP.REF	Map references
011	ZONE	Zones
012	STOP.SEATS	Additional seated passengers (+s,-s)
013	STOP.WHCH	Additional wheelchair passengers (+w,-w)
014	X.Y	Map reference x,y co-ordinates
015	PASS.TRIP.NO	Passenger Trip references
016	TRIP.TYPE	Trip types (O/C/B)
017	MO.FLAG	Multi-occupancy flag
018	FARE	Fares (For drop-off events)

Single-valued attributes:

019	VEH.TRIP NO	Vehicle Trip No.
020	CANCELLED	Cancellation flag (Y/N)
021	CANC.DATE	Cancellation Date
022	VEHICLE NO	Vehicle No.

Multi-valued - 1 value per driver:

023	DRIVER NO	Driver Nos (Driver shifts are not necessarily coterminous with vehicle shifts)
024	DRIVER.START	Driver start times

Multi-valued - 1 value per driver break

025	BREAK.NO	No. of driver break
026	BREAK.START	Break start time
027	BREAK.LENGTH	Length of break in minutes

Multi-valued - 1 value per assistant

028	ASSISTANT NO	Assistant No
029	ASSISTANT START	Assistant start time
030	ASSISTANT FINISH	Assistant finish time

Shift records are created by the appropriate data entry program, and may be cancelled and eventually archived.

The only subroutine which changes shift entries is the scheduling subroutine SCHEDULE for Dial-a-Ride or SC SCHEDULE for Social Car, described in H 7 k below.

H. Programs

1. Program Types

Within MULTI TRIP, programs are of the three types mentioned in section C.2 above and, in addition, DATA/Basic programs which have been written to perform functions which are not available in System Builder. Furthermore, many programs in MULTI TRIP are catalogued external subroutines, which can be called from System Builder programs, from generated programs or from other DATA/Basic programs or subroutines

Some functions are also performed by PROCs. These are mainly complex ACCESS statements.

2. Flow of Control

Control may be passed between programs, subroutines and PROCs in a number of ways.

a) CALL

A program or subroutine may call a catalogued subroutine. A subroutine is catalogued with the statement:

```
CATALOG filename subname
```

and an entry created in the Master Dictionary of the account.

Within a program, a call is made either by calling the subroutine by name, or by setting a variable to the name of the subroutine and using CALL @·

```
CALL subname

var = subname
CALL @var
```

Parameters may be passed to these subroutines if they are called by other DATA/Basic programs. Parameters are passed by value and returned in the same way.

```
CALL subname (param1,param2,param3,  )
```

A called subroutine begins with the keyword SUBROUTINE, followed by the name of the subroutine and a list of parameters in brackets

```
SUBROUTINE subname (param1,param2,param3,...)
```

Alternatively, called subroutines have access to the common block of the calling program, and data can be passed using the common block In System Builder generated programs, the common block is coded in as lines of the program In subroutines and programs written before the involvement of Wise Software Ltd., the common block was held as a separate item in each program file, and included with the INCLUDE compiler directive

```
INCLUDE COMMON BLOCK
```

In this way, changes to the System Builder common block in subsequent releases of System Builder could be made easily, and programs and subroutines recompiled.

A called subroutine must terminate with a RETURN statement

b) CHAIN

Control can also be passed from one program to another program (not a subroutine) by the use of the CHAIN command. Any TCL statement can be chained, but it is most often used to transfer to another program in a suite of programs. If the program is catalogued, the following statement will work:

```
CHAIN "progname"
```

otherwise:

```
CHAIN "RUN filename progname"
```

The use of the I option to RUN prevents initialisation of variables in the chained program and allows data to be passed in a common block:

```
CHAIN "RUN filename progname (I"
```

Control is not passed back to the chaining program.

c) EXECUTE

A DATA/Basic program may execute any statement which could be run from TCL. A new workspace is initialised, and the process runs in the new workspace. Workspaces may be pre-initialised on PC PICK systems with the :TASKINIT command in SYSPROG. This should be added to the USER-COLDSTART PROC:

```
H:TASKINIT 10,9
P
```

This initialises 10 workspaces, with a limit of 9 levels of EXECUTE per user. The command should not be used while users are on the system. EXECUTE is similar to a shell in DOS or UNIX, but a new prompt is not provided, simply the ability to run a process in a new workspace, leaving the original process intact. The use of EXECUTE carries a disk overhead of about 200k per workspace. If workspaces are not pre-initialised but the individual user limit is not exceeded, there is a slight delay while workspce is allocated from the overflow table and linked, each time an EXECUTE statement occurs in a program. Exceeding the user limit is fatal to all processes for that user!!

When an executed process terminates, control returns to the program which exceuted the process. Screen output from the executed process may optionally be captured to a variable, and PICK error numbers can be returned in a variable.

```
EXECUTE "TCL statement" CAPTURING var1 RETURNING var2
```

The statement may be assigned to a variable and then executed:

```
var="TCL statement"
EXECUTE var
```

EXECUTE was not available on many early PICK systems, and complicated techniques were developed whereby programs wrote PROCs which were then chained, and which terminated by passing control back to the original program by running it.

Data may be passed to input statements in an executed program or PROC using the DATA statement, allowing programs to bypass prompts in the executed process, e.g :

```
DATA 1,'Y','0034'
EXECUTE "DISPLAY.DRIVER"
```

Executed processes cannot pass data back to the original program, except by writing data to temporary file items.

Warning. In PC PICK, and in some other releases of PICK which lock groups of items in files rather than individual items, the termination of an executed DATA/Basic program will release all locks for that user!

System Builder provides a means of running any parameter driven program using the slash '/' at any prompt. This leads to the execution of the drivers for the appropriate type of program, and an overhead of 2 executes for most processes, and 3 executes for reports.

The same result can be achieved from within a program or subroutine by executing "SB" and using the DATA statement to pass the name of the parameter driven program to it.

3. Program Files

All MULTI TRIP program files are in the account MTPROGS, with Q pointers to them from the MULTITRIP account

They are:

CTPROGS	-	System Builder generated programs
SUBPROGS	-	Common and Group Hire programs and subroutines
DRPROGS	-	Dial-a-Ride programs and subroutines
SCPROGS	-	Social Car programs and suborutines
MTPROCS	-	PROC file
WSLUFO	-	Wise Software utilities
RW.BP	-	Report-Wise
UPGRADE	-	Programs to upgrade MULTITRIP account

Other utility files may be found in a development version of the system

All System Builder programs are in the account SYSTEM.BUILDER (referenced as SB4.2 on some systems). They are in the files:

```
DM
DMAD
DMGS
DMOT
DMUTIL
```

Object code only is provided by System Builder

4 System Builder Menu Structure

The System Builder Menu System provides the main structure for MULTI TRIP When a user logs to the MULTITRIP account, the System Builder program DM00 normally runs, and prompts the user for a password If a valid password is entered, the user is taken through a MULTI TRIP opening screen into the Main Menu, which is provided by the System Builder program MM. The first menu to be displayed is the item MAINMENU. All the menus are held in the file CTCONTROL.

The complete menu hierarchy for MULTI TRIP is attached as Appendix 1.

Each attribute in a menu item represents one option on the menu, and values within each attribute represent the information required in order to perform that option The important information is as follows:

```
MENU PARAMETER ITEM STRUCTURE

VALUE    USE
-------------------------------------------------------------
1        FILENAME (BLANK IF MENU)
2        MENU ENTRY
3        NAME OF PARAMETER ITEM
4        SYSTEM BUILDER DRIVER
            DM20.DRIVER - DATA ENTRY
            DM21.DRIVER - TRANSACTION
            DM22.DRIVER - ENQUIRY
            DM23 - AUTOMATED PROCESS
            DM24 - UPDATE PROCESS
            DM25 - BATCH TRANSACTION
            DM26 - REPORT
            DM27 - STANDARD LETTER
   OR
         NON-SB PROGRAM NAME

5        TYPE
            P - PROGRAM
            M - MENU
            S - SUBROUTINE

6        SUB-MENU OPTIONS
7        PROGRAM RUN BEFORE MAIN PROGRAM
10       GENERATED PROGRAM NAME
11       PROGRAM RUN AFTER MAIN PROGRAM
12       PRIVILEGE LEVEL REQUIRED
```

5 PROCs

The following PROCs are in the file MTPROCS

DRIVER LISTS

Builds lists of drivers for each service and of all drivers, both as select lists in POINTER-FILE and as items in CTLISTS, where the list name or id is 'D*' followed by the service code, as used in the file GPSERV.CODE

DRIVS

Calls DRIVER.LISTS. Should be copied to the Master Dictionary of the MULTITRIP account.

VEHICLE.LISTS

Builds lists of vehicles for each service and of all vehicles, both as select lists in POINTER-FILE and as items in CTLISTS, where the list name or id is the service prefix.

VEHS

Calls VEHICLE LISTS. Should be copied to Master Dictionary of MULTITRIP account.

Other PROCs in MTPROCS are utility PROCs

6. Programs and Subroutines

Appendix 2 contains details of every program or subroutine in the files listed in Section H.3 above. For each program, any comments in the first 14 lines of the program are shown, followed by details of every transfer of control by CALL, EXECUTE or CHAIN to another subroutine, program or process In the case of a CALL, the parameters in the CALL statement are matched up to the parameter list in the SUBROUTINE statement at the start of the called subroutine. A list of all System Builder subroutines used by each program or subroutine is also provided. For subroutines, as well as the above, the parameter list at the start of the program is given.

7. Key Functions

The menu structure in Appendix 1 and the list of programs and subroutines in Appendix 2 provide documentation of the MULTI TRIP package, but do not provide the depth necessary for an understanding of the key functions of the system These are described in the section below.

a) Set Up System Parameters

A file, CTSYSTEM PARAMETERS, with a Q pointer to it as CTSP, contains details of the organisation using MULTI TRIP, and of aspects of its working practices which determine the way in which certain parts of MULTI TRIP will perform important functions. The item MT contains these details. It is maintained by a menu option in the Common Module Set-Up Options menu. This process is run in parameter driven form by System Builder, as it involves potentially 8 screens, more than can be generated into a single DATA/Basic program.

The first two screens contain general details, and subsequent screens apply to each service. These facilities would be better divided up into separate programs for each module in a future version

The contents of the item MT in CTSP are copied to the item MTDATA in CTCONTROL by the program WRITE MTDATA which runs after the Set-Up System Parameters screens. The item DATA in CTCONTROL must contain the following two values in attribute 4:

004 MTDATA]20

The System Builder subroutine SET COMMON, which is called by all generated programs and should be called by non-SB programs will copy the system parameters into element 20 of the PARMS array in the common block, so that they are accessible to all programs and subroutines. This facility was not available in System Builder Releases 3.1 or 4.1, and is not used by all programs in the Group Hire and Common modules which were originally written in SB3.1.

If Field 15 in Screen 2, "Generate Headings", is set to 'Y' then a program, GEN HEADING, runs after the Member Group record screen, and copies details from the name and address and Company No., VAT No., etc. to produce a heading for documents, which it is therefore assumed will not be printed on headed stationery.

The name and address of the organisation using MULTI TRIP is held in the Mailing List file, CTMAILING, in item 0000 A Member Group record for the organisation is held as item 0000 in the Member groups file, GPMEMBER

b) Create a Vehicle Booking

Bookings for the use of vehicles are made by:

Own Use Bookings
Group Hire Bookings
Dial-a-Ride Shifts
Social Car Schedules

In each case, there is a System Builder screen program to enter the data associated with the booking, and a System Builder program to enter details of Regular Bookings or Shifts. Regular Bookings are dealt with in the next section

For Own Use Bookings and Group Hire Bookings, a Booking may consists of more than one Vehicle Trip: a booking may consist of a combination of outward, continuation, return and round trips for more than one vehicle. For Dial-a-Ride and Social Car Schemes, a Shift or Schedule consists of a single Trip. It is the vehicle trip which is checked for feasibility as described below.

The following screen programs are used to enter the necessary details to create bookings and shifts:

Group Hire	I*GPBOOKINGS*BOOKING1
Own Use	I*CTBOOKINGS*BOOKING1
Dial-a-Ride	I*DRSHIFTS*SHIFT1
Social Car	I*SCSHIFTS*SHIFT1

A new version of the Group Hire booking screen has been produced, which is under test, I*GPBOOKINGS*BOOKNEW1. There is also a version of the Group Hire screen specifically for Vehicle Brokerage, I*GPBOOKINGS*BROKING1, which links into the facility to search for a suitable brokerage vehicle described in sub-section (g) below

When a booking is made, a number of checks are made on the vehicle chosen. For Group Hire, these are initiated by the subroutine CHECK.GP.VEHICLE, which validates the entry of the vehicle number

Check vehicle is in service
Check vehicle is available for Group Hire service
Check there is not another trip or trips in the diary which clashes with the one being made (GP.CHECK.DIARY)
Check vehicle availability against weekly pattern of usage by owning group, e.g. vehicle is not available Mondays or on Tuesday mornings (CHECK AVAILABILITY).
Check vehicle has facilities required by hirer, e g tail-lift or tow-bar (VEH.CHECK.FAC).
Check vehicle is capable of carrying required combination of seated and wheelchair passengers (VEH.CHECK.PASS).

Some of these checks may be overridden by the operator

For Own Use, the subroutine is CHECK.CT.VEHICLE, and only the diary and availability checks are made (CT CHECK DIARY and CHECK AVAILABILITY)

For Dial-a-Ride, the subroutine CHECK DR VEHICLE is used, and this performs the following checks.

Check vehicle is in service.
Check vehicle is available for Dial-a-Ride service.
Check there is not another trip or trips in the diary which clashes with the one being made (DR.CHECK DIARY)
Check vehicle availability against weekly pattern of usage by owning group, e.g. vehicle is not

available Mondays or on Tuesday mornings (CHECK.AVAILABILITY)

For Social Car service, the subroutine CHECK.SC.VEHICLE is used, and this performs the following checks

Check vehicle is in service.
Check vehicle is available for Social Car service.
Check there is not another trip or trips in the diary which clashes with the one being made (SC.CHECK.DIARY).
Check vehicle availability against weekly pattern of usage by owning group, e.g. vehicle is not available Mondays or on Tuesday mornings (CHECK AVAILABILITY).

In each case, the letters TBAL (TO BE ALLOCATED LATER), may be entered instead of a vehicle number, in which case a check is simply made on the number of TBAL trips already entered in the diary for that date or those dates.

If errors are detected by any of the checks, it is possible to pursue various options. The most significant of these is the one to view the Vehicle Diary, described in sub-section (f) below.

If a booking or shift is acceptable, the diary (CTDIARY) is updated, and entries in the vehicle trips file (CTTRIPS) are made when the booking or shift is filed at the end of the data entry screens. See sub-sections (d) and (e) below.

c) Create Regular Bookings

For each service it is possible to create regular vehicle bookings from which ordinary vehicle bookings can be generated automatically (as set in system parameters) Regular bookings are created in data entry screens which are similar to those for the creation of ordinary bookings. However, they do not include data items for dates, instead they include a set of data items which define the frequency of the regular booking. For each service, the screen programs are the following

Group Hire	I*GPREGULARS*BOOKING1
Own Use	I*CTREGULARS*BOOKING1
Dial-a-Ride	I*DRSHIFT REGULARS*SHIFT1
Social Car	I*SCSHIFT REGULARS*SHIFT1

The data items which define the frequency of the ordinary bookings generated from the regular bookings are as follows for Group Hire

```
PERIOD          Over what period does the booking repeat?
                Months/Weeks/Days (M/W/D)
FREQUENCY       Every how many months, weeks or days does it
                repeat? (1/2/3/4)
DAY.OF.WEEK     If it repeats monthly or weekly, on what day of the
                week is it? (1-7)
WEEK.OF MTH     If it repeats monthly, does it repeat on 1st, 2nd,
                3rd, 4th or last week of month? (1-4/L)
START.DATE      When is first booking to occur?
FINISH DATE     When is last booking to occur (inclusive)?  May be null
REVIEW DATE     When should booking be reviewed?  Mandatory if
                FINISH.DATE is null.
```

These data items allow bookings to be set up for every day, every Friday, every first Tuesday of every second month, etc

In Group Hire, after the screen has been completed a subroutine REG BOOK1 is called which allows a hire agreement to be printed, and the user may choose to create another similar booking for the same group A subroutine REG BOOK2 is called after the file has been updated, which deals with actually prınting the hıre agreement and setting up a repeat booking. It also calls a subroutine GEN.BOOKINGS with the key of the regular booking which has just been created as its single parameter GEN BOOKINGS reads the regular booking from the file and generates ordinary bookıngs as appropnate, e g every Fnday, from the start date untıl the lımıt for bookıng ahead in the dıary for the servıce (as set up ın system parameters), e g up to 2 months ahead. It checks the diary for clashes, vehicles not available, etc. and wıll create bookings allocated to TBAL rather than a specıfic vehıcle, and produces a report of bookings whıch have had to be dealt wıth in thıs way

For other Own Use the subroutines are

```
REG.BOOK1
CT.REG.BOOK2
CT.GEN.BOOKINGS
```

For Dial-a-Ride and Socıal Car services, shıfts (which are vehicle bookings in these services) repeat weekly. Instead of the shift key containing the actual date of the shift, it contains a number representıng the day of the week on which ıt repeats (1-7). The ordinary shıfts are generated up until the limit for booking ahead for the servıce. In the case of Dial-a-Ride, this ıs done by the subroutine DR.REGULAR.SHIFT; ın the case of Social Car by SC.REGULAR SHIFT. Checks are made for dıary clashes, shıfts may be allocated to TBAL ıf they clash wıth other bookıngs, and a report is printed out detailing changes made.

The generatıon of ordınary bookıngs or shıfts from regular bookings or shifts ıs simplıfied by the use of PICK internal date format, number of days sınce 31st December 1967, and by the use of PICK date conversıon functıons, whıch allow, for example the numeric day of the week of a particular date to be derived by a statement such as.

variable = OCONV(expressıon,conversıon)

DAY = OCONV(DATE(),'DW') (gives numenc day of week of today's date)

Once a regular bookıng has been set up, subsequent generatıon of bookıngs ıs controlled by values ın the system parameters One value determınes whether the diary should be updated daıly when the system is fırst used For each servıce, ınformatıon ıs held about the penod ahead for which bookings may be taken, and this is used to determıne whether to update the diary for that servıce, and how far ahead bookings should be generated. Values are also held in the system parameters whıch ındicate how far ahead bookıngs have already been generated, ın case the book ahead parameters are changed

Typically, for Group Hıre, the book ahead unıts will be months and the period will be 3 or 6 (range is 1-12, default ıs 6). Bookıngs wıll then be generated monthly for up to 4 or 7 months ahead, allowıng for regular bookings to have been generated up to 3 or 6 months ahead throughout the whole month wıhout the need to run the generatıon routine on a daıly basis, and preventing ordinary bookıngs from being taken before regular bookıngs have been allocated to their usual vehicle

Typıcally, for Dıal-a-Rıde, the book ahead unıts wıll be days, and the penod only 1 or 2 Shıfts wıll then be generated daıly for 1 or 2 days ahead

The unıts may be days, weeks or months, and the range of the period ıs validated in conȷunction wıth the unıt

Own Use Bookings are booked as far ahead as the maxımum of the other servıces set up on the system.

Updatıng of the bookıngs fıles and dıary from regular bookıngs is performed by the subroutıne MTDIARY.UPD, whıch is called by the startup routine ıf the system parameters are set for automatic update. MTDIARY UPD calls a separate subroutine for each service, if that service is installed. These are:

```
GP DIARY UPD
CT.DIARY UPD
DR.DIARY.UPD
SC.DIARY.UPD
```

These subroutines are similar to those which run after the screen program which sets up the regular booking in the first place, but all interaction with the user is eliminated in order to allow the process to run unattended Reports of changes will be printed out.

The subroutine MTDIARY.UPD may also be run as a menu option in the Vehicle Diary menu, DIARYMENU, which is usually used if automatic update is set off.

d) Create a Vehicle Trip

Each vehicle shift in Dial-a-Ride and Social Car services is a single vehicle trip; a Group Hire or an Own Use booking may consist of more than one trip. For each such trip, an entry is made in the vehicle trips file CTTRIPS The trip is the basic unit of vehicle use.

For Group Hire, this and the creation of an entry in the diary for each trip are done by the subroutine CHECK BOOK, which calls GPTRIPS UPD to create the actual trips If a booking has been amended, GPTRIPS.UPD will compare the old version on file with the version in memory and delete the old trips and create new ones Each trip has a 7 digit key including a check-digit, using modulo 11.

For Own Use, CTCHECK BOOK calls CTTRIPS UPD. For Dial-a-Ride, DR.SHIFT.CHECK calls DR.TRIPS UPD For Social Car, SC SHIFT.CHECK calls SC.TRIPS UPD.

When created, the trip items contain the following attributes·

```
001 Booking Number/Shift Key
002 Trip type (O/B/R/C)  Out/Back/Round/Continuation.
003 Start Time
004 Finish Time
005 Vehicle Number
006 Driver Number
007 Number of Passengers
008 Number of Wheelchairs
009 Start Date
010 Finish Date
011 Service (O/G/D/S)
012 User Number/Service
```

e) Check Diary

All checks on actual vehicle availability are made by calls to the subroutine DIARY in SUBPROGS. DIARY is also the only subroutine or program in the system which updates the diary file, CTDIARY

DIARY must be called with the following parameters:

```
MODE      Type of action required
          1 - Enter new trip and release locks or return error
          2 - Check for insertion of trip, return error if
              not possible, leave vehicle-day locked if no
              error (new trip should be inserted very soon)
          3 - Delete and leave locked (for subsequent insertion
              of different trip)
          4 - Delete and release
DATE      Date of Trip
VEH       Vehicle No.
BOOK.NO   Booking No. or Shift No.
JOUR.NO   Journey or Trip No.
START     Start time
FINISH    Finish time
TYPE      G,D,S,O
ORIG      Group or Service if D-a-R or SC
```

A value is returned in

```
INTERNAL.ERROR
          0 - Okay
          1 - Another trip starts at same time
          2 - Another trip overlaps
          3 - Another trip finishes at same time
          4 - NOT USED
          5 - Trip to be deleted cannot be found
          6 - Trips allocated to TBAL have been counted
              and value is in PARMS(17)
```

The Vehicle No. can be TBAL (To Be Allocated Later), which allows bookings to be taken in brokerage, or to be allocated later, or to be de-allocated from one vehicle in order to be reallocated.

The diary checking algorithm uses the LOCATE function in DATA/Basic, the format of which is:

LOCATE(expression,arrayvar{,attrexp{,valexp}};setvar{,sequence}) {THEN statement{s}}
ELSE statement{s}

It evaluates expression, attrexp (attribute expression = attribute number in dynamic array) and valexp (value expression = value number in attribute in dynamic array). It then attempts to find the result of evaluating expression in the dynamic array, a particular attribute (if attrexp is present) or in a particular value (if valexp is present) It sets setvar equal to the position of the attribute, value or sub-value in the array if it is found, or to the value of the attribute, value or sub-value at which it should be inserted. This last only has meaning if sequence is specified. Sequence can be 'AR', 'AL', 'DR' or 'DL' for ascending or descending right or left Left is used for strings to be positioned by ASCII value, and right is used for numeric values to be positioned in numeric sequence. If no sequence is specified, setvar defaults to the last element of the array. If expression has been found in the array, the THEN statement(s) are executed, otherwise the ELSE statements are executed.

For a new trip to be inserted in the diary, its start-time must not be the same as the start-time of another trip, its finish-time should not be the same as the finish-time of another trip, and the location in the array of start-times at which both its start-time and finish-time would be inserted must be the same, and the location in the array of finish-times at which both its finish-time and start-time would be inserted must be the same. The use of PICK internal format for times (seconds since midnight), the use of the 'AR' sequence, and the use of the LOCATE function provide a fast check on the diary

The subroutine DIARY only updates a single vehicle day at a time. The responsibility for splitting trips which run over more than one day, as they can in group hire, rests with the calling program or subroutine For example in the group hire booking screen, this is done by GP.CHECK DIARY. In group hire, other subroutines which call DIARY are GP.INS.DIARY and GP.DEL.DIARY, which are used after a booking has been entered or amended. The subroutine GP.REL.DIARY releases locks on vehicle-days in CTDIARY. DIARY is also called directly by GEN.BOOKINGS which generates bookings from regular bookings, and by AMEND.BOOKINGS, which carries out limited amendments to generated bookings.

In Dial-a-Ride, DIARY is called by DR.INS.DIARY, DR REGULAR.SHIFT (which also calls DR.INS.DIARY), DR.SHIFT.CHECK (which also calls DR.INS.DIARY) and DR.CHECK.DIARY.

In Social Car Scheme, DIARY is called by SC.INS.DIARY, SC.REGULAR.SHIFT (which also calls SC.INS.DIARY), SC SHIFT.CHECK (which also calls SC.INS.DIARY) and SC.CHECK.DIARY.

In the Common module, four subroutines which may be run as part of the start-up routine, and which update the Bookings/Shifts, Trips and Diary files with Bookings or Shifts based on Regular Bookings or Shifts, call DIARY. These are GP DIARY.UPD, CT.DIARY.UPD, DR.DIARY.UPD, SC DIARY UPD Each one also calls the XX INS DIARY subroutine for the appropriate service

Also in the Common module are subroutines relating to Own Use Bookings These are CT.CHECK.DIARY, CT INS.DIARY, CT.DEL DIARY, CT.GEN BOOKINGS, CT.AMEND BOOKINGS

f) View Diary

The vehicle diary acts as the main check on actual vehicle availability, there are three programs which provide views of the diary and allow the user to navigate through the diary and perform many vehicle booking, shift entry and post-trip data entry functions which would otherwise only be available through the menu system

The main view of the diary shows a single vehicle day, and shows details of each trip on that vehicle day This is a System Builder generated enquiry screen, O*CTDIARY*DIARY, which has been amended considerably to provide the necessary functionality. The current vehicle day is effectively seen through a window onto an array of vehicle days, with days running vertically and vehicles running horizontally and wrapping round. The user may use the numeric keypad to navigate through the diary

```
4 - move left, i e. to the previous vehicle
6 - move right, i.e. to the next vehicle
8 - move up, i.e. to the previous day
2 - move down, i e. to the next day
7 - move back a week
1 - move forwards a week
9 - move back a month
3 - move forwards a month
5 - 'zoom-in'
```

When the user 'zooms-in' control is transferred to the subroutine ZOOM.IN, and a highlighting cursor is displayed on the first trip in the diary for the day. This can be moved up and down the list of trips, using the numeric keypad (8/2) When a trip is highlighted, a number of options are available:

View the booking/shift for that trip,
Amend the booking for that trip;
Make the post-trip entry for that trip;
Move the trip (group hire only).

If the trip is moved, the user can then move to another vehicle day in the diary, select 'zoom-in' and insert the trip on that vehicle day All the same checks are made as are made when a booking is created or amended, as described in sub-section (b) above A trip selected to be moved can be cleared from memory.

When 'zoomed-in' the user can also create a new booking or shift or regular booking or shift for any service.

It is this view of the diary provided by O*CTDIARY*DIARY that the user gets if he or she checks the diary from a booking or shift entry screen. It can in fact be accessed by the user entering '/D' at any System Builder prompt

There are two other views of the diary, equivalent to one another, which can be accessed from the menu system, and which can pass control to the level described above. These are provided by the programs DIARY SUMMARY and DIARY SHEET. In each, several vehicle days can be viewed on the screen in an array

```
DIARY.SUMMARY
     5 vehicles
     3 days
     4 lines per vehicle day
     12 characters per line
     each character represents 30 minutes of vehicle use

DIARY SHEET
     10 vehicles
     14 days
     1 line per vehicle day
     4 characters per line
     each character represents 6 hours of vehicle use
```

More vehicles can be displayed on 132 column screens.

In each case, each character is either a dash indicating no use of the vehicle at that time, or a letter indicating the service for which the vehicle is in use at that time (O/G/D/S)

In each case, the top left vehicle day is highlighted on entry to the screen, either by reverse video or a different colour this highlighting cursor can be moved around to different vehicle days, using the numeric keypad as in the more detailed diary. The options to move forwards and back a week or a month redisplay the screen having moved to the new date. In addition, the whole screen can be shifted right or left or to the start or end of the list of vehicles. The TBAL dummy vehicle is shown at the end of the list of vehicles, and there is no wrap round.

By moving the highlighting cursor to a particular vehicle day, and pressing 5 to 'zoom-in', the user can view the detailed diary screen described above.

The diary routines maake extensive use of the EXECUTE statement, and adequate workspaces should be pre-allocated

g) Search for a Suitable Vehicle

When a Vehicle Brokerage booking is made, it is necessary to identify a suitable vehicle or vehicles for the trip(s) By enteiing 'FIND' as the Vehicle No. in the booking entry screen (I*GPBOOKINGS*BROKING1), the subroutine FIND.GP.VEHICLE is called by CHECK.VB.VEHICLE in order to identify suitable vehicles.

The subroutine performs two types of process
Vehicles are sorted in an order which reflects the order in which they should be considered;
Vehicles are checked for their availability and the suitability of their facilities.

The sort uses two sets of data:

The distance of the vehicle base from the start point of the trip;

The vehicle sequence number, which is a measure of how "difficult to book" a vehicle is.

This sequence number is designed to reflect the number of constraints on a vehicle A vehicle with many constraints, such as limited facilities, extensive use by the owning group, limited client groups which it can carry (for example only elderly people), should be considered for bookings early on, and vehicles which are easy to book should be considered only when difficult to book vehicles have already been booked or considered and rejected The sequence number is a value in the range 0-99 held on the vehicle record, the larger the number, the more difficult to book

The distance is calculated dynamically, based on the map reference of the start point of the trip and the map reference of the vehicle base held on the vehicle record

The PICK ACCESS enquiry language is the fastest way of performing a sort in PICK. The map reference of the start point is converted to a numeric set of coordinates and written to an item on a file with a key which ieflects the port on which the process is running. A correlative in the data dictionary calculates the distance using Pythagoras In fact, the distance is left as the square of the distance, as there is no facility in ACCESS correlatives to take square roots, but the effect is the same.

Having sorted the vehicles available for Group Hire by distance and by sequence number, each vehicle is then considered in turn. The same checks are applied as when checking vehicle availability to make a booking If a vehicle is suitable, it is added to a list of suitable vehicles. If not, it is added to a list of unsuitable vehicles, with the reason for its unsuitability, e g. another booking at the same time, or does not have a tail-lift

The user may then view and scroll through either list, and can choose a vehicle from the list of suitable vehicles, whose vehicle number will be returned to the data entry screen field.

h) Sort Pick-up Addresses for Group Hire

The Travelling salesperson problem applies to the sorting of passenger addresses into a suitable pick-up order Traditionally in the travelling salesperson problem, the distance between points is known and forms the basis of the heuiistic used In Group Hire, it may be necessary to pick up passengers from a number of addresses, for which the distance between each address and each other address is not known, and could only be calculates with a detailed mapping database and route-finding software or estimated using giid references The approach used here is based on the space-filling curve heuristic developed by Bartholdi et al. in Atlanta, Georgia, and uses a mapping of position in a unit square to position on a unit curve Any grid refernce can be mapped to a point on the curve, and this is done using an ACCESS correlative to a look-up table. The addresses are then sorted by their position on the unit curve, and this is used as the basis for the pick-up sequence.

This process is perfomed by the subroutine GP.PASSLIST.REP which executes an ACCESS statement, which uses the exploding sort connective BY-EXP to explode multi-values out of an attribute, which is not available in System Builder reports Up to four passenger lists from the file GPPASSLIST can be merged and sorted The output is sent to the printer.

i) Enter Post-Trip Details

After a vehicle trip has been completed, data from the vehicle log-sheet or from the driver's worksheet can be added to the trip record. Once this has happened, the trip can no longer be deleted and replaced or amended by amending the booking or shift.

The post-trip details may be entered in either of two ways:

in a single data entry screen per trip;

in a transaction screen laid out like a vehicle log-sheet with
several trips to a screen (CTLOGTRANS)

The assumption is that trip data is entered in time sequence, as the latest mileage reading in the vehicle record is updated, although this is not essential. The latest mileage reading is used by a report which predicts likely service dates based on the last service date and mileage, the current date and mileage, and the service interval for the vehicle (VEH.EXPIRIES).

The existing attributes may be amended to reflect the actual vehicle, driver, start time, etc. Additional are added to reflect what has taken place. The start odometer reading is retrieved from the vehicle record for the vehicle concerned, but may be amended. The finish odometer reading is entered, and the distance calculated.

A driver expenses record (CTEXPENSES), a vehicle damage record (CTDAMAGE) or a vehicle fuel record (CTFUEL) may be linked to the trip record and entered at the same time For Group Hire, these will be used when the cost of the booking to which the trip belongs is calculated for invoicing purposes, e.g to charge for driver expenses and damage to the vehicle and to give credit for fuel bought by the hirer.

j) Create a Passenger Booking

There are screen programs for creating passenger bookings in Dial-a-Ride and in Social Car service. The Social Car scheme booking screen is currently run as an interpreted parameter-based program (SCBOOKINGS BOOKING1), while the Dial-a-Ride screen is a generated program, I*DRBOOKINGS*BOOK1

The two programs are very similar, differing mainly only in fields such as Driver No. which is present in Social Car, as the choice of driver is important, and in the types of other passengers which are present in Dial-a-Ride where they may be significant.

The bottom right hand corner of the screen (from row 15 column 37 to row 22 column 76) is kept clear of prompts and is used to display various information and look-ups which otherwise would require clearing and refreshing the screen

The Social Car screen also uses some look-ups on indexes which reflect critical user comments on the standard System Builder look-up facilities, and which emulate facilities available in System Builder 5 1 (SUBPROGS B LOOKUP)

In both, the passenger number is entered, and details of the passenger are displayed. The date of the trip(s) is then entered. Trips may be of three types. Outward, Continuation and Back (O/C/B) If the first trip is outward, the passenger's home address, zone and map reference are used as defaults. A list of the passenger's regular destinations is displayed and one can be chosen by number. A landmark destination (3 character mnemonics for regular destinations of all services, e.g. LHR - London Heathrow) may be entered An address may simply be entered. An address may be entered and added to the passenger's list of regular destinations.

Map references must be used if the trip distance is to be calculated automatically and the fare is dependent on distance. Zones must be entered if the fare is on a zone-to-zone basis.

The pick-up time is entered, and the estimated distance is calculated from the map references, using the grid size and 'wiggle factor' in the system set-up parameters. The wiggle factor is a multiplier to apply to the pythagorean distance between the grid centres to reflect the density of the road network. The estimated drop-off time is displayed as a default. This is calculated from the estimated distance, using the average speed and trip constant in the system set-up parameters. The trip constant reflects the time taken to get a passenger onto and off a vehicle

The Shift must be entered, either as a shift number such as '1D' or as 'TBAL'. A look-up of available shifts for that day is provided. When a shift is entered, the subroutine CHECK.DR.SHIFT calls the shift scheduling subroutine SCHEDULE for Dial-a-Ride or SC.SCHEDULE for Social Car, to check whether the trip pick-up and drop-off can be inserted into the shift. This is described in sub-section (k) below. These facilities allow viewing of shifts and other bookings

Details of the passenger's disability, aids such as wheelchairs and special requirements. Requirements are held in a code file with an additional field to reflect the level of requirement. For example, a low step is '2', while a tail-lift is '3'. The value of the passenger's requirement code must be less than or equal to the maximum facility code for the vehicle. A passenger who requires a tail-lift cannot be accommodated on a vehicle with only a low step, but a passenger who requires a low step can be accommodated on a vehicle with a tail-lift Group hire requirements such as tow-bars and roof-racks are set to '0'.

Details of the number of other people being carried with the passenger are entered, and may be used for the purpose of calculating the fare The fare is calculated by DR.FARE.CALC, or SC.FARE.CALC for Social Car It uses details held in DRCHARGES or SCCHARGES and potentially in DRZONE.ZONE or SCZONE ZONE, depending on how the fare is calculated. It may be based on distance or on zone to zone rates.

For a Return or Continuation tiip, most of the details default from the outward trip

After all trips have been entered in a booking, a summary is displayed in the bottom right hand corner of the screen. The user is asked whether the booking is accepted, and only if they answer 'N' do they see the second screen, which prompts for refusal or cancellation details.

The initials of the user who made the booking, the booking date and the start and finish times for the booking are all recorded This is designed to provide a means of monitoring the amount of time spent on bookings rather than pioviding a check on staff.

In the Social Car scheme booking screen slightly different facilities are available for selecting a shift it should be noted that the same data structure has been used for both Dial-a-Ride and Social Car, based on the fact that one customer was using the Dial-a-Ride module for Social Car trips. This has subsequently proved to be unsatisfactory for some users, and further thought should be given to this module Probably the data structure is adequate but the idea of setting up shifts in advance needs to be reconsidered It is important, however, that the concept of distinguishing between passenger trips and vehicle trips is retained, and that passenger trips in a Social Car scheme can be overlapped, whereas vehicle trips cannot.

The Driver field is in the Social Car screen, as it may be that the passenger asks for specific driver. if the passenger has a prefeired driver then the subroutine DRIVER.DEF fetches the driver number and displays it as default. If the passenger has more than one preferred driver, then their names are displayed and the user can select one by number.

If the user enters 'FIND' the subroutine FIND.SC.DRIVER is called to check all the Social Car drivers for availability and to display them in two lists, suitable and unsuitable drivers, with the distance of the location of their vehicle from the start point of the trip

When the driver has been selected, his or her details are displayed, and the details of the vehicle he or she owns If 'S' is entered in the Shift field, details of shifts for the vehicle owned by the driver in the Driver field are displayed If a vehicle number is entered in the Shift field, a list of any existing shifts for that vehicle on the day of the trip(s) is displayed.

When a Shift Number is entered, the same facilities are available for checking the shift as in Dial-a-Ride

k) Schedule a Passenger Trip

Entering a Shift Number and Letter (e g. 1E) in the Dial-a-Ride or Social car Booking screens will check the existence of the shift and whether the trip times are within the range of the shift start and finish times. This is done by the subroutine SCHEDULE for Dial-a-Ride and SC.SCHEDULE for Social Car

These subroutines work in a similar way to the DIARY subroutine, except that they have to check for two events to be added into the shift, as both pick-up and drop-off have to be inserted in sequence. Whereas vehicle trips cannot overlap in time, passenger trips can, and indeed must if efficient use of vehicles is to be made Also, it may be necessary to have more than pick-up and/or drop-off at the same time, for example when several passengers are travelling to or from the same address at the same time

The scheduling subroutines are called with the following parameters

MODE	Mode of use (1/2/3/4) 1 - Enter or return error code 2 - Check and return error, leave locked if no error 3 - Delete and keep locked 4 - Delete and release locks
DATE	Trip date
VEHICLE.NO	No of vehicle or shift
SHIFT.LETTER	Letter for shift - typically 'D' Day, 'E' Early or 'L'
BOOK NO	Booking No.
THIS.TRIP.NO	No of Passenger Trip
START	Start time (pick-up time)
FINISH	Finish time (drop-off time)
TYPE	Type of trip (O/C/B)
WAIT	Wait for passenger, i.e. drop and wait. Not valid for trip type 'B', back home.
O.MAP.REF	Map reference of origin
D.MAP.REF	Map reference of destination
O.ZONE	Zone of origin
D.ZONE	Zone of destination
O.X.Y	Numeric map reference as x,y grid co-ordinates for origin
D.X.Y	Numeric map reference as x,y grid co-ordinates for destination
O.ADDRESS	Address of origin
D.ADDRESS	Address of destination
PASS	No of passengers
WC	No. of passengers in wheelchairs
MOO.FLAG	Multi-occupancy origin flag, indicates multiple pick-ups at same time at origin are acceptable
MOD FLAG	Multi-occupancy destination flag, indicates multiple drop-

```
                 offs at same time at destination are acceptable
FARE             Fare for trip
```

A value is returned in

```
INTERNAL.ERROR Error code returned

            1    Shift does not exist
            2    Times of trip not in shift range
            3    Another stop at the same time
            4    Another drop-off at the same time
            5    Old trip pick-up not in shift (when trying to delete)
            6    Old trip drop-off not in shift (when trying to delete)
            7    Old trip pick-up and drop-off not in shift (when trying
                 to delete)
            8    Number of existing TBAL pick-ups etc. in PARMS(17)
            9    Pick-up clashes with driver break start or finish
            10   Drop-off clashes with driver break start or finish
            11   Trip overlaps driver break
            12   Trip during driver break
            99   Shift record locked by another user
```

Whether there is space or not, a message is displayed, offering the user the opprtunity to view the shift record. This is provided by DR.SHIFT.SCHEDULE for Dial-a-Ride and SC.SHIFT.SCHEDULE for Social Car This routine displays various details in a window made up of three sub-windows The top window displays the date, shift no and shift letter and start and finish times. The user can move between different shifts using the numeric keypad. For example if shift 1E, where E is for early, is displayed, pressing '6' will move to shift 2E, if it exists. Pressing '2' when in shift 1E will move to 1L, where L is for late, if it exists. Details of the shifts for a day are held in the shifts index file, DRSHIFTS.INDEX, in a record keyed on the internal format of the date Pressing <return> accepts the allocation to the shift currently displayed. Pressing <escape> returns to the shift field without accepting the allocation. As each shift is displayed in the topmost window, details of 5 pick-ups and drop-offs in that shift are displayed in the bottom window, with the pick-up or drop-off nearest the time of the trip being scheduled in the middle of the window

Pressing '5' in the numeric keypad moves the user into the middle window, where the time of the pick-up or drop-off for this passenger trip are displayed with the event type, address, name and map reference . Pressing '2' or '8' switches between pickup and drop-off. Pressing <return> or <escape> moves back up to the top window Pressing '+' or '-' adjusts the time of the currently displayed event in 5 minute intervals Pressing <escape> moves back to the top window and disregards changes made to times using '+' and '-', whereas <return> accepts these changes. Pressing '5' moves the user into the bottom window and highlights the middle line of the window. Each line shows the time of the event, the type, the address, the booking no. and the map reference. The display can be scrolled up and down using '2' and '8' to go one event later or earlier, '3' and '9' to go 5 events later or earlier, and '1' and '7' to go to the end and start of all the events in the shift. Finally, entering '5' shows details of the booking which is currently highlighted, using O*DRBOOKINGS*SMALLBOOK, a System Builder generated program which has been changed so that it does not clear the entire screen.

I. Training and Tutorials

Two tutorials, developed using System Builder provide an introduction to the facilities of MULTI TRIP and the keyboard conventions of System Builder, using the Mailing List.