MONASH UNIVERSITY
THESIS ACCEPTED IN SATISFACTION OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
ON........................ 25 March 2003 ....

....
Sec. Research Graduate School Committee

Constraint-Based Software for Broadband Networks Planning

- A Software Framework for Planning with The Holistic Approach -

Afwarman Manaf

M.Sc.

A thesis submitted for the degree of
Doctor of Philosophy
in the
Department of Electrical and Computer Systems Engineering,
Monash University, Clayton,
Victoria, Australia

February 2000

Contents

List of figures v

List of tables viii

Abstract ix

Declaration xii

Acknowledgements xiii

Acronyms and Abbreviations xv

Chapter 1. Introduction 1

1.1 What is the hardest problem? 2

1.2 Constraint-based software 3

1.3 Research goals 7

1.4 Conclusion 8

Chapter 2. Broadband Network Planning 11

2.1 Broadband networks 12

2.2 Broadband network technologies /services 14

2.3 Hierarchical planning 15

2.3.1 Customers and services demand forecast 16

2.3.2 Service layer 16

2.3.3 Network layer 17

2.3.4 Costing, revenue and profitability layers 19

2.3.5 Feedback 19

2.4 Optimality criteria 20

2.5 Metropolitan area network planning 21

2.6 Intercampus network planning 23

2.7 ISDN acces network planning 24

2.8 Software for automated planning of broadband networks. 25

2.9 Conclusion 26

Chapter 3. Approaches to Planning 27

3.1 Broadband network planning problems 27

3.2 Existing approaches 30

3.3 Structuring the planning knowledge 34

3.4 Constraints knowledge 36

3.5 Objects 39

3.6 Research focus 41

3.6.1 Integrated planning system architecture 43

3.6.2 Incorporating different types of knowledge into objects 44

3.6.3 Case-base as means to preserve experiences 45

3.7 Conclusion 46

Chapter 4. Constraint-Based Planning System 47

4.1 Planning system architecture 47

4.1.1 Planning scenario 49

4.1.2 Gaining efficiency 50

4.2 Constraint-based planning 51

4.3 Processing the constraint network 54

4.4 Deductive model for constraint propagation 55

4.5 Planning rules 58

4.6 Backward propagation of current results 60

4.7 Case-base module 62

4.8 Database support 62

4.9 Human planner role 63

4.10 Conclusion 63

Chapter 5. Constraint Knowledge **64**

5.1 Input data as first constraints 65

5.2 Procedural algorithms 67

5.2.1 Generic planning algorithm 67

5.2.2 Specialized algorithm 72

5.3 Mathematical models 73

5.3.1 Predicting traffic from all customers 73

5.3.2 Estimating traffic from the network services 75

5.3.3 Estimating the network profit 77

5.4 Rules of thumb 78

5.5 Past cases make the solution-finding simpler 81

5.6 Constraints from the database retrieval result. 82

5.7 Conclusion 82

Chapter 6. Objects for Constraint Propagation **84**

6.1 Objects 84

6.1.1 Attributes, functions and operations 85

6.1.2 Object class 86

6.1.3 Inheritance hierarchy 87

6.1.4 Relationships between objects 90

6.1.5 Planning knowledge within objects methods 92

6.2 Process control and data flow 94

6.3 Input data for planning process 96

6.4 Propagating planning constraints through objects 94

6.4.1 Deductive process 103

6.4.2. Abductive process 104

6.4.3. Evaluation of possible solutions 110

6.5 Conclusion 112

Chapter 7. Evaluation and Conclusion **114**

7.1 Design analysis 114

7.1.1 Modeling the planning entities 114

7.1.2 Constraint-based model 116

7.1.3 Constraint knowledge 117

7.2 Planning solution 117

7.3 Computation time 118

7.4 Conclusion 121

7.4.1 Further work 123

Appendix-A. Snapshots from the planning system 124

Appendix-B. Software developed for the prototype system 132

Appendix-C. Published papers 152

References 180

List of figures

Figure 1.1	Basic schema of the planning system	5
Figure 2.1	Broadband network	12
Figure 2.2	Hierarchical planning layers	15
Figure 2.3	Metropolitan Area Network (MAN)	22
Figure 2.4	Intercampus network	23
Figure 2.5	ISDN network	24
Figure 4.1	Planning system architecture	48
Figure 4.2	Constraint network	51
Figure 4.3	More detailed constraint network	53
Figure 4.4	Backtrack search	54
Figure 4.5	Deductive model of access network technology options	56
Figure 4.6	Deductive model of backbone network technology options	56
Figure 4.7	Constraint propagation using the deductive model	57
Figure 4.8	Rules for access network technology selection	59
Figure 4.9	Rules for intercampus network technology selection	59
Figure 4.10	Guiding the search with rules	60
Figure 4.11	Backward propagation	61

Figure 5.1	The planned area	65
Figure 5.2	Existing cables system	66
Figure 5.3	Network planning steps	68
Figure 5.4	Planned area customers distribution	75
Figure 5.5	Technology migration rule	79
Figure 6.1	Object structure	85
Figure 6.2	Network plan object class	86
Figure 6.3	Network plan object internal code	87
Figure 6.4	Planned area subclasses	88
Figure 6.5	Inheritance hierarchy of 'service' object	88
Figure 6.6	'Service' object and its inheritance objects	89
Figure 6.7	Hierarchical structure of 'network plan' object	89
Figure 6.8	Object classes and relationships	90
Figure 6.9	An object sending a 'traffic load' message	91
Figure 6.10	Sending a message of the 'preferred technology'	92
Figure 6.11	A method containing a complex mathematical formulae	93
Figure 6.12	A method containing a rule of thumb	93
Figure 6.13	Control flow of the planning process	94
Figure 6.14	Data flow of the planning process	96
Figure 6.15	Broadband network planning input data	96
Figure 6.16	Provided services in a certain area	97
Figure 6.17	Broadband network service demands	98
Figure 6.18	Service characteristics	99
Figure 6.19	Constraint propagation over the network of objects	100
Figure 6.20	Technology plan object	101
Figure 6.21	MAN technology options	102

Figure 6.22 Propagating a traffic load constraint 103

Figure 6.23 Generating possible solutions 104

Figure 6.24 Reduced possible solutions for the network technology 106

Figure 6.25 DQDB-MAN requirement 108

Figure 6.26 Equipment data example 108

Figure 6.27 Backward propagation of partial results 111

Figure 7.1 Hierarchical telecommunications networks as an object 116

Figure 7.2 Computation times of the planning processes 119

List of tables

Table 3.1 The proposed system compared to the others 42

Table 5.1 Service demands 70

Table 5.2 Services characteristics 75

Table 5.3 Videophone service demand data 76

Table 5.4 MAN switching system data 82

Table 6.1 MAN planning input data 99

Table 6.2 Intercampus network planning input data 99

Table 6.3 Intercampus network technology options 102

Table 6.4 MAN planning solution 112

Table 6.5 Intercampus network planning solution 112

Abstract

Planning broadband networks with a holistic approach is a very difficult task. Existing planning softwares are not capable of addressing the task in an integrated manner. Most of them only address a specific sub-task and leave the rest to be dealt with separately. This causes inefficiency and can increase the possibility of inconsistency in the network planning process. This thesis describes a new approach to realise a software that provides a framework to address broadband network planning problems in a more comprehensive way.

The thesis proposes a *constraint-based model* as the core of the software architecture. The *constraint-based model* is a description of significant planning objects and relationships amongst them. The model incorporates all the *constraints* internal to each object and also the *constraints* of each object in relation to other objects. The planning process begins with the creation of the internal model in a certain initial state. The planning process can be seen as the process of model refinement according to input data and *constraints*. Any input data and *constraint* may lead to discrepancy with the model. This discrepancy forces the model to be *refined* in order to meet consistency. The model refinement process changes the

current variable value of the objects into a new value which is consistent with the values of the other variables.

The thesis describes the design and implementation of a prototype planning software to find solutions for the task of broadband network planning. It describes a deductive process that takes place in the *constraint-based model* and also describes different types of *abductive knowledge* which are integrated into the model to realise a viable planning system. *Abductive knowledge* contains planning rules, which might be subjective or may change according to different circumstances, and analytical methods which have a more solid scientific foundation. The planning system also includes case-base and database modules to realise an effective system for solving network planning problems efficiently. Last but not least, the human planner as the user is also a part of the system. The planner can 'guess' or select the plan by imposing his or her preferences if the system knowledge does not lead to the solution which is satisfactory to them.

This work attempts to achieve at least four research goals. The first goal is to design an architecture for general planning systems, and implement it for the broadband network planning problem. The second goal is to build an innovative application of *artificial intelligence*, particularly *constraint-based reasoning* for carrying out broadband network planning tasks. The third goal is to build a novel planning software which has a unified mechanism to find solutions by enforcing consistency, together with the use of multiple sources of knowledge to reduce the size of the search space. And the fourth goal is to design and implement a kind of *knowledge-based system* with multiple types of knowledge by incorporating knowledge into a *network of objects*.

The superiority of the proposed planning software lies in the *constraint-based model* which is successfully combined with all the other components namely, analytical methods, rules of thumb, case-base, and database as well as the involvement of the human planner.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other institution and, to the best of the author's knowledge, this thesis contains no material previously published or written by another person, except where due reference is made in the text of this thesis.

[redacted]

Afwarman Manaf

Acknowledgements

It is my pleasure to acknowledge the many people involved through the completion of this thesis. My supervisors, Dr. Bruce Tonkin, Dr. Leith Campbell and Dr. Kishor Dabke, provided many hours, insight, discussion and encouragement during our regular meetings. Dr Kishor Dabke helped me since the beginning of my Ph.D candidature, advised and corrected me in writing this thesis. He continued all of these as a honorary co-supervisor upon his retirement. His support and help were always there when I needed them.

Dr. Leith Campbell helped me to understand the other side of the issues I was tackling. He provided useful information and also patiently helped and corrected me in writing this thesis. Dr. Bruce Tonkin acted as my main supervisor upon Dr. Dabke's retirement. His role as my main supervisor at the latter stage of my Ph.D candidature was found to be very useful and therefore was invaluable for the completion of this thesis. He provided support, advice and much encouragement.

I especially want to convey my gratefulness to Dr. Masahiro Takatsuka. He helped me to set up my computer system for this research work when he was in this department as a Ph.D student. I would like to thank Mr. Farzad Safaei for the discussions and Dr. Nigel Aylott for useful information. Both of them are from Telstra Research Labs. I also want to thank Mr. Jeremy Ginger and many other people in the Optus Design Centre, Sydney, Australia. I am also grateful to Mr. Suryatin Setiawan for allowing me to spend some time working in the Information Technology Research Centre, P.T. Telekomunikasi Indonesia, Bandung, Indonesia.

I would like to thank the chairman, faculty members and staff, and the numerous other students in the Department of Electrical and Computer Systems Engineering, Monash University. I also thank all the good people in ANSPAG/CTIE. All of them helped me to enjoy my time as a Ph.D student.

Finally, I dedicate this thesis to my wife, Indrakasih, for her sacrifice and constant support; and to our children, Rahmi and Umar, who have missed much of their 'play time' with me since I had to stay much longer in front of my computer to complete this thesis.

Acronyms and Abbreviations

AI	Artificial Intelligence
ATM	Asynchronous Transfer Mode
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAN	Customer Access Network
CBR	Continuous Bit Rate, or Constant Bit Rate
CGW	Customer Gateway
CHT	Call Holding Time
CPE	Customer Premises Equipment
CSP	Constraint Satisfaction Problem
BHCA	Busy Hour Call Average
BISDN	Broadband Integrated Services Digital Network
DBMS	Data Base Management System
DQDB	Distributed Queue Dual Bus
EGW	Edge Gateway
FDDI	Fiber Distributed Data Interface
FR	Frame Relay

ICI	Inter Carrier Interface
IEEE	Institute of Electrical and Electronics Engineers Inc.
IMG	Inter MSS Gateway
ISDN	Integrated Services Digital Network
LAN	Local Area Network
MAN	Metropolitan Area Network
MSS	MAN Switching System
NISDN	Narrowband Integrated Services Digital Network
NMC	Network Management Centre
OR	Operations Research
PABX	Private Automatic Branch Exchange
POTS	Plain Old Telephone Service
PSTN	Public Switched Telephone Network
SMDS	Switched Multimegabit Data Service
SR	Subnetwork Router
VBR	Variable Bit Rate
WAN	Wide Area Network

Chapter 1

Introduction

Freedom is one of the most basic properties of life, but constraints are always there. It is not critical to ask what is acceptable for someone to do because one basically can do everything so long as it does not violate any constraint. The amount of knowledge about what is acceptable is too big and an attempt to know all of them is a futile attempt. On the other hand, the knowledge about all the constraints is highly critical, and an attempt to know all the constraints perhaps is the most intelligent attempt. Indeed, real life is a matter of constraints.

This work uses the above mentioned simple notion to deal with telecommunications networks, particularly broadband network planning problems. This thesis builds a software for planning broadband networks in a more comprehensive way.

Broadband network planning features the involvement of different sorts of constraints. Carrying out the planning tasks involves technical, social, economic and even political factors. These constraints are not only technical but also non-technical. A broadband network plan solution should reflect what is permitted, what is restricted, and what is mandated by many factors which interact throughout the planning process.

1.1 What is the hardest problem?

Broadband network planning is defined as a process which involves methods and techniques for planning the overall network. The objective of the planning is to produce specifications of the network technology and the pieces of equipment to serve the broadband communications demand in a certain planned area [1]. Planning processes range from the selection of the network technology to the dimensioning of the network equipment to meet predefined objectives e.g. to maximise *business profitability* [2].

Searching for a planning solution is the hardest problem. The planning process finds the solutions for the network which should satisfy both technical and economical constraints. Moreover, the telecommunications network is a rapidly changing environment. The inter-relations of evolving services and technologies to existing infrastructures and access markets also implies the need for elaborate techno-economic analysis [3].

Broadband network planning has been carried out in a fragmented manner because planning broadband networks with a holistic approach is a very difficult task. For example, transmission systems planners often deal only with the issues of transmission loss and digital coding by assuming that the underlying physical infrastructure of fibres, wires and cables is reliable. External network planners simplify planning problems to a set of planning rules by neglecting considerations of the overall network performance and cost. Most of the planners address a specific sub-task and leave the rest to be dealt with separately.

This separation has been necessary because of the limitations of the architecture and performance of the existing softwares. The softwares which adopt microscopic approaches have caused inefficiency and possible inconsistencies. A new software architecture is required to support the planners to carry out such a highly complex task in a more comprehensive manner [4].

The availability of a comprehensive software for planning broadband networks has recently become a necessity, and a holistic approach toward broadband network planning is the most appropriate approach. This is due to a trend towards increased technical complexity and diversity of network infrastructure plans. This is also due to the ever increasing need to understand the detailed interactions of all factors involved in the network planning process.

Although the development of a software for integrated network planning is very difficult, the framework for such system which deals with the problems with a holistic approach is highly desirable.

1.2 Constraint-based software

This thesis adopts the notion of planning as a process of finding a solution to a problem which is composed of a finite set of variables. Each variable is associated with a certain domain. There is a set of constraints that restricts the values the variables can simultaneously take. The task is to assign a value to each variable satisfying all the *constraints* [5].

This thesis proposes a new approach by developing a *constraint-based model* as a central component of the software architecture for a better planning software. This model is constructed by taking advantage of the knowledge about the structure of the planning entities, and applying such knowledge as constraints in making planning solutions.

The *constraint-based model* is the core of the system which contains a description of all planning entities and implements all the relationships amongst them. This model incorporates all the *constraints* internal to each entity and also those of each entity in relation to other entities. These *constraints* include all pieces of *planning knowledge* that can be expressed in the form of mathematical relations, rules of thumb, specific optimisation methods, or procedural algorithms. The planning system uses such different types of *knowledge* as sets of *constraints* to be satisfied in the internal process.

A basic schema of an overall planning system architecture is shown in Figure 1.1. The planning system consists of two major parts, namely *constraint-based model* and *abductive knowledge*. The two parts differ in the way they direct the changes of the variable values throughout the planning:

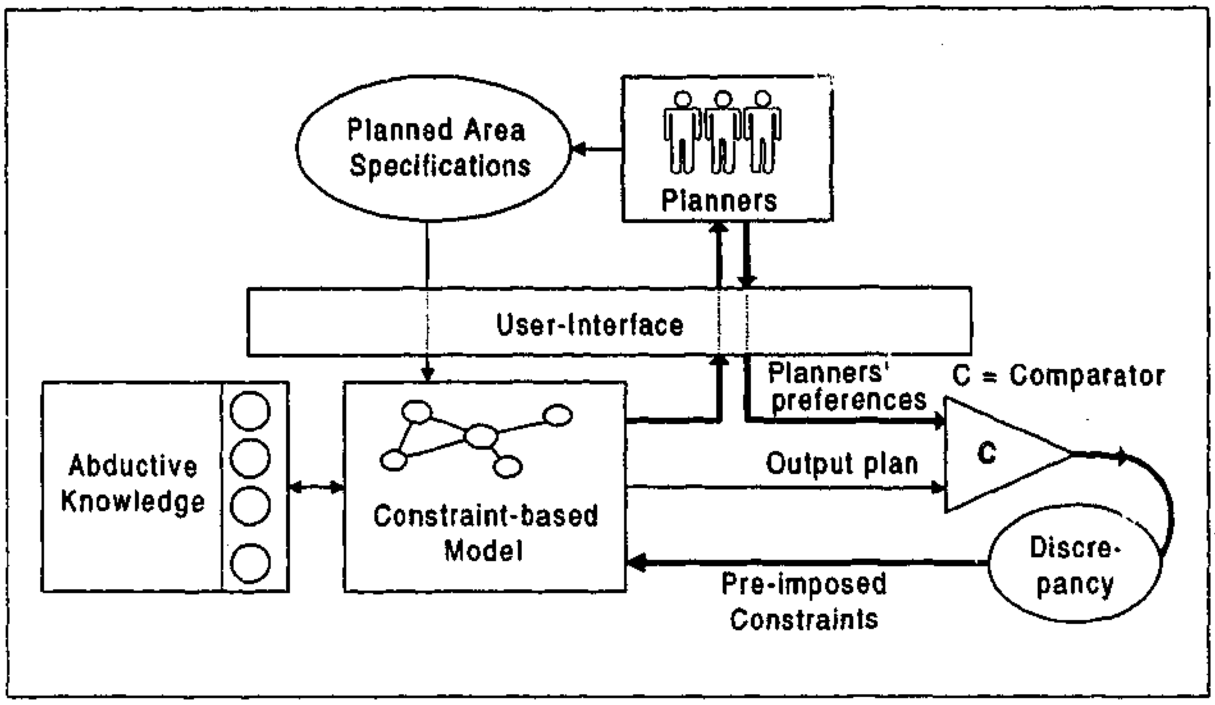


Figure 1.1: Basic schema of the planning system

- The *deductive process* uses *constraints* and propagates them over the *constraint-based model* to draw conclusions. It uses a *deductive model* which is designed so that it guarantees at least one possible compound value would match the plan requirements. The model is capable of covering all possible values of the network plan entities. The *deductive model* provides the 'freedom' of choosing any solution.
- The *abductive process* uses planning *knowledge* and specifications or input data to assign some known values to the planning entities. This *abductive knowledge* puts *constraints* on choosing any solution. It helps the system to quickly assign values to the planning variables. However, the *knowledge* is very limited and it is unlikely to be complete.

Abductive knowledge includes planning rules, which might be subjective or changing according to different circumstances, and analytical methods, which have more solid scientific foundation. The *knowledge* from past experience and information retrieved from a database is also considered as *abductive knowledge*. The *abductive knowledge* is instrumental to realise an effective system for solving broadband network planning problems quickly. The preference of the human planner (user) is also seen as an *abductive part* of the system's *knowledge*. He or she can also 'guess' or select a plan by imposing their preferences if the *knowledge* incorporated within the system does not produce the solution which is satisfactory to them.

Abductive knowledge is all the planning *knowledge* to direct, guide and simplify the *deductive process*. It is used to deal with the combinatorial explosion, and makes the approach of this thesis significantly superior to other approaches. *Abductive knowledge* guides the search and eliminates the unlikely alternatives so that it avoids unnecessary exhaustive searches within the *deductive model*. All types of *abductive knowledge* support the *constraint-based model* to find solutions very efficiently.

The *planned area specifications* in figure 1.1 include all descriptions of the planned area and the data necessary for commencing a planning process. These specifications are the first *constraints* for deducing any plan solution. They are usually formulated by the human planners.

The *discrepancy* in figure 1.1 is any inconsistency that occurs between the deduced plan and the *constraints* from the human planners' preferences. This discrepancy will trigger the *deductive process* to take place again to form an iterative cycle to accommodate such pre-imposed *constraints*. It drives the deduction process to find another solution as illustrated in Figure 1.1.

Actually, the cycle also takes place internally in the *constraint-based model* before yielding any output plan. This approach is very powerful and it can be generalised to address the planning problems in various kinds of domain. However, it can generate too many logically possible alternatives. In the worst case it has a complexity of $O(k^n)$ [6], where k is the average size of the variable domain which is assumed to be finite, and n is the number of variables involved. Therefore, this thesis proposes an integration of *abductive knowledge* into the *constraint-based model* to reduce the complexity of the search space.

It is important to note that only one plan is considered at one time in the iterative process shown in Figure 1.1. However, the planning system is capable of generating more than one solution which meets the specifications (input data) and the human planners' preferences.

1.3 Research goals

There has been an increasing demand for an integrated software for broadband network planning [33]. This work attempts to achieve the following research goals. The first goal is to design a new software architecture for general planning systems, and implement it for a broadband network automated planning software.

The second goal is to build an innovative application of artificial intelligence, particularly *constraint satisfaction techniques* [18] for carrying out broadband network planning tasks.

The third goal is to build a novel planning software which uses a unified mechanism for accessing multiple types of *knowledge* to reduce the complexity of the search within the solution space. The proposed system reduces the severity of a typical problem which occurs in hybrid systems, namely the problems of integration and coordination amongst different modules. The fourth goal is to design and implement a software with different types of *knowledge* by incorporating them into *objects* [47]. This goal is achieved by performing a problem analysis of broadband network planning; system modeling; and by designing and implementing a *network of objects* as a computational engine for the broadband network planning software.

1.4 Conclusion

This chapter has introduced a new approach for building softwares to carry out broadband network planning tasks. The system uses a *constraint-based model* supported by multiple types of planning *knowledge* to tackle a very difficult broadband network planning problem. The new approach is proposed to ensure the efficiency of the solution finding.

The chapter has identified the hardest problem of the broadband network planning task for which this thesis attempts to solve. The chapter has explained a basic schema of the software architecture and has briefly discussed the *constraint-based model* and the *deductive process* taking place within the model. It has also described different types of *abductive knowledge* used by the planning system.

The chapter has briefly shown how to integrate all of the system components to make the proposed system superior to other existing planning software. At the end, this chapter has also pointed to the motivation behind the thesis and some research goals to be achieved in the thesis project.

The rest of the thesis is organised as follows. The next chapter, chapter 2, discusses the broadband network planning problems in greater detail. It elaborates hierarchical planning layers used in this thesis to deal with the extreme complexity of the planning problems. It also describes planning problems of Metropolitan Area Network (MAN), Intercampus network and ISDN access networks.

Chapter 3 discussses existing approaches to planning and elaborates the research problem by reference to the literature. This chapter points out the remaining problems for research work and emphasises the focus of this thesis. Chapter 4 describes the overall planning system architecture for broadband networks. This chapter discusses all components of the system in greater detail.

Chapter 5 discusses all pieces of the planning *knowledge* for finding solutions. These *knowledge* items are viewed as *constraints* which are very important for directing the solution finding process. Chapter 6 explains the use of *objects* as the means for incorporating all the *knowledge* required. This chapter describes a method for processing a constraint network in terms of propagating *constraints* over the *network of objects*. Chapter 6 also describes a step-by-step process of the planning system.

Chapter 7 evaluates some important aspects and output of the planning system. It also discusses the computational efficiency of the software. At the end, this chapter provides a conclusion and suggests some points for further research.

Chapter 2

Broadband Network Planning

The process of planning a broadband network is more an art than a science. There are some salient features of the problem including [7]:

- It is a problem from wide-ranging disciplines. A holistic approach to the problem is more desirable than a microscopic approach.
- It is a *semi-structured problem*. Domain knowledge or expertise plays a very important role for solving the problem.
- It is a highly complex problem. Problem complexity increases exponentially as the scope and size of the problem enlarges.
- It is a constraint-intensive problem. Many requirements are often imposed and should be met by the resulting network, such as requirements of network interworking, throughput, response time, reliability, cost, maintenance and performance.
- It is a problem of a rapidly-changing environment. The problem solutions can be very dynamic. Therefore, a satisfactory solution is often pursued as opposed to the 'best' solution.

Broadband network planning is the development of a network infrastructure with existing technology, meeting services demands which have inherent uncertainty, introducing new network technology with new products, and improving network facilities. A broadband network planning solution should meet corporate needs, business and customers' requirements. Furthermore, the planning process must recognise economy-of-scale and time-horizon when synthesising costs of network equipment [8].

2.1 Broadband networks

A broadband network is a telecommunications network that supports multimedia applications where voice, data, image and video traffic are integrated. Broadband network service is defined as a service that requires transmission channels capable of supporting rates greater than 1.5 Mbps or primary rate ISDN or T1 or DS1 in digital technology [9]. A broadband network can be divided into two main parts, namely access and backbone networks, as shown in Figure 2.1 [1].

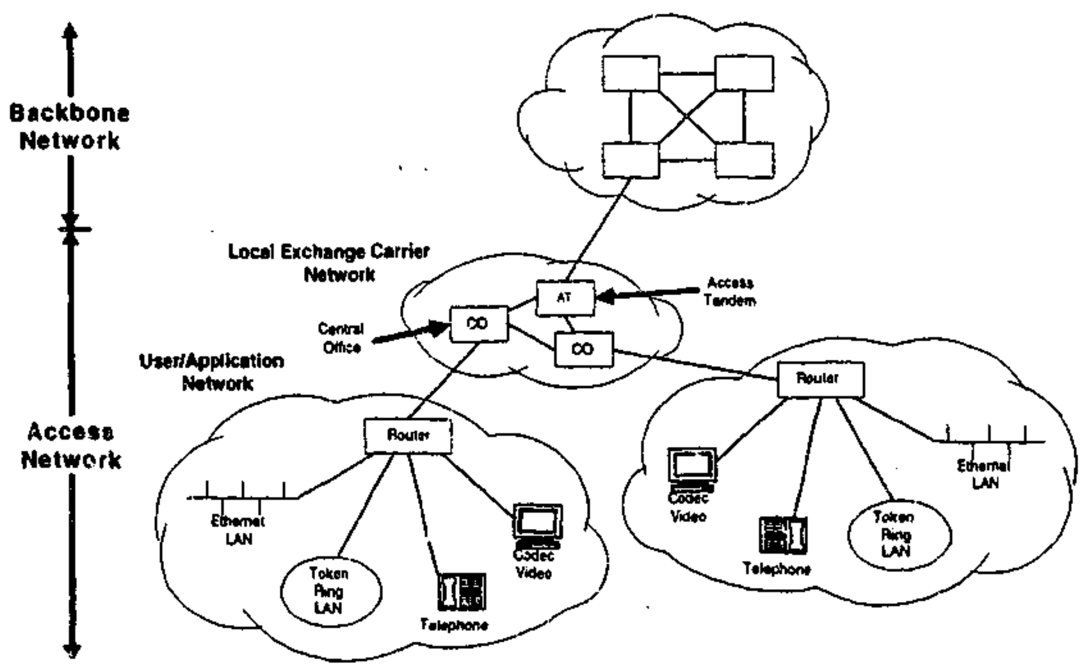


Figure 2.1: Broadband network

An access network is defined, in Figure 2.1, as a network layer below the backbone network. It is the network where all traffic is originated and terminated. It consists of a user/application network and a local exchange carrier network. User/application network is where the users are connected to the network or where the user accesses the network applications. The local exchange carrier network consists of central offices which cover a certain local region and an access tandem that aggregates the traffic. The access tandem connects one access network with the others. The backbone network is the network above the access network and it is also called inter-exchange carrier network [1].

Broadband network planning deals with three major categories of components, namely customer premises equipment (CPE), switching, and transmission links. CPE is located at the user's premises for transmitting and receiving user information. CPE also exchanges control information with the network. The CPE can be a telephone, computer terminal or video codec facility. Switching systems or nodes interconnect the transmission links at various locations and route the traffic through the network. Transmission links provides communications paths to carry the customer traffic and network control information between nodes in a network.

This thesis project develops a software to assist the network planners in the process of planning broadband networks. Therefore, it deals with some important issues regarding capacity expansion and some planning issues. The issues include estimating the traffic load, selecting the network technology, and sizing transmission links, nodes or switching. The process should cover categories of network requirements such as: traffic, protocol, architecture, network services, features and functions. However, the software does not provide traffic simulation or emphasise the modeling of broadband network traffic characteristics.

2.2 Broadband network technologies/services

There is a variety of network technologies available for network applications including fibre distributed data interface (FDDI), distributed queue dual bus (DQDB) network, narrowband integrated services digital network (NISDN), Frame Relay, switched multimegabit data service (SMDS), asynchronous transfer mode (ATM) network, and broadband ISDN (BISDN). FDDI and DQDB both use optical fibre as the transmission medium and are targeted for LAN interconnect application. FDDI is primarily intended for a backbone network to interconnect private LANs and a version called FDDI II is capable of carrying both isochronous (voice) and data traffic. FDDI typically runs at a rate of 100 Mbps with effective throughput of 80 Mbps [10].

DQDB is intended for public networks and is also capable of carrying both voice and data traffic. It supports both circuit-switched and packet-switched services with data rates varying from 34 Mbps to 155 Mbps. DQDB can utilise many types of transmission systems. DQDB and FDDI are both categorised as access technologies. Frame relay (FR) supports transmission over a connection-oriented path with access speeds of 56 Kbps or multiples of 64 Kbps. However, it has capacity limits of the highest rates at 1.544 Mbps (T1) or 2.048 Mbps (E1 for Europe). FR is most suitable for LANs interconnect although it can support client-server computing, CAD/CAM application, graphic application or information systems application.

ATM is a network technology considered as an ultimate solution. It guarantees successful transport of any service (CBR = constant bit rate or VBR = variable bit rate). It also has unlimited growth path for higher speed rates and B-ISDN.

2.3 Hierarchical planning

Hierarchical planning is defined as a method of dividing a complex network planning problem into layers of subproblems. Figure 2.2 depicts such a hierarchy. This figure adopts a fairly similar concept of layering as found in [11].

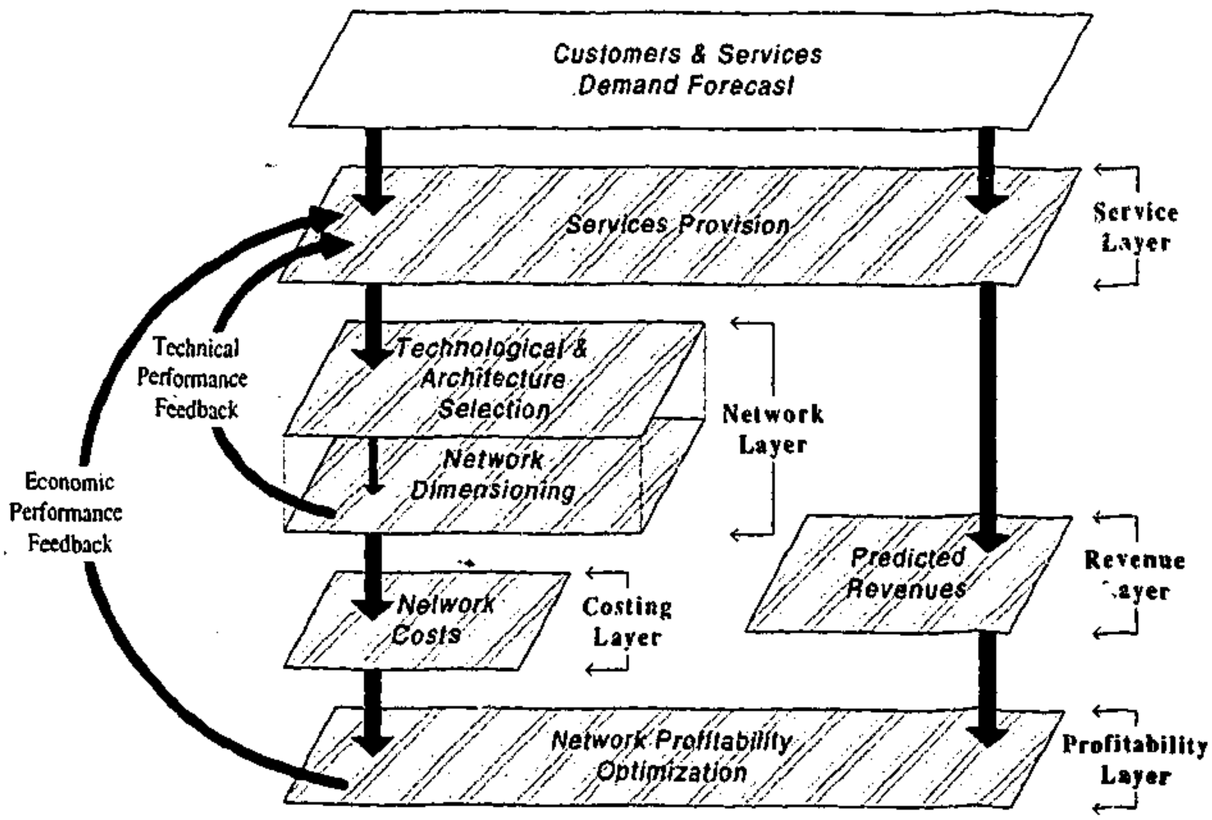


Figure 2.2: Hierarchical planning layers

2.3.1 Customers and services demand forecast

This layer is an input layer to the planning process. It categorises customers in each area to be planned and models the services for forecasting demand. A methodology for modelling customers, services and traffic has been described in [12]. Analysis of customers and services aims to determine user needs and the services suitable for them. This layer foresees services penetration, traffic volume and growth.

This layer provides a traffic load prediction in a certain area to be planned. Traffic load analysis uses the parameters of Penetration (P) of each service for the planned area, Busy Hour Call Average (BHCA) and Call Holding Time (CHT). The results of the analysis includes:

- Traffic load of different customer classifications;
- Total traffic load from different regions;
- Inter-region traffic demand.

This layer determines the values of the above parameters for each service and customer group.

2.3.2 Service layer

This layer is heavily based on the customers and services demand forecast. It determines the types of services to be provided so that it can also be called a service provision layer. This layer deals with most of the uncertainty and complexity due to the nature of the forecast. The uncertainty mainly comes from forecasts based data about services penetration. Each service penetration scenario has a probability factor associated with it.

This layer embodies many inter-related factors mentioned at the beginning of this chapter. Of course, other factors such as existing services and available network technology and all future technology under development should be considered when planning the services provision.

2.3.3 Network layer

The output of the service provision layer feeds into the network layer in terms of parameters that can be used for network planning e.g. service requirement, bit rate, and quality of services. The network layer consists of switching and transmission modules to assemble a network which delivers such parameters.

This layer selects an appropriate technology and architecture for the network which has a big impact on overall network plans. New technologies should enable the improvement of current services and creation of new services. This process considers all implications of each technology decision of the network plan. Planners should have an understanding about technology developments and use the knowledge about existing networks and constraints.

Selecting the topology of the network is also part of this layer. This process determines the topology of the backbone and access networks, and it usually appears only in long term planning. It is performed to face an increasing demand of transmission media capacity. This process relies heavily on the experience of planners and the following factors should be considered [13]:

- existing infrastructure;
- traffic distribution;
- required access facilities for customers;
- functional characteristics of the network elements.

Network dimensioning determines a network configuration to satisfy the demands of each network site, which is based on the topology as well as required capacities [14]. It results in a network configuration with the specifications of types of the equipment and quantities required. The network configuration should be selected to satisfy the demand and it is determined by the results of:

- network topology selection;
- required bandwidth calculation;
- network elements specifications and costs.

The analysis of the existing infrastructure is essential in planning the network. Network planning is based on existing infrastructure which includes both access area and backbone area infrastructure. This process analyses existing network infrastructure such as the current network technology, the central office location, the existing transmission lines, and the transmission system.

2.3.4 Costing, revenue and profitability layers

The optimality criterion used in this planning process is maximum profitability. Profit is represented by all potential revenues generated, from which costs are subtracted. However, besides the economic criterion, a network planning should certainly meet all technical requirements to meet services demand.

The costing layer uses the network layer's output to apply economic criteria for network costs analysis. This layer should calculate total costs and business overheads of all services to be provisioned. The revenue layer basically takes the customer base information together with services usage to compute the predicted revenues which are calculated based on predefined tariff information. The profitability layer is the last and the most decisive layer for evaluating a network plan. It takes the outputs of total costs and predicted revenues and applies a series of commercial accountability calculations such as profit/loss, balance sheet, and cash flow.

2.3.5 Feedback

Feedback is very important for finding an optimal network planning solution. The planning system uses a financial performance indicator of the current solution as feedback for further optimising the network plan. It analyses the economic performance of the network from different feasible configurations, which satisfy the given traffic load and certain quality of service criteria. The feedback process forms an iterative process for finding an optimum solution.

2.4. Optimality criteria

A complete network plan consists of both the technological and configuration plans and gives the optimal solution in terms of network profitability [11]. This process uses total costs of the network plan and predicted revenues from the network to determine optimality. Total cost consists of equipment, upgrade, management and installation labour costs as well as lifetime operational costs. Fixed and variable costs are all considered as well as the cost for complete and partial roll out for different technology options. At the same time, this process also predicts generated revenues from the network. Predicted revenues are based on several possible market penetration scenarios from different market survey models provided to the planners. Therefore, a suitable method to find optimal plans is needed to deal with market uncertainty as well as complexity of the solution alternatives.

Network planning aims to compromise between maximum short term and long term profitability. Planners should be able to create a sustainable plan in terms of profitability rather than merely gaining maximum short term profit which may result in big losses in the future. Uncertainty of the market penetrations and rapid changes of the network technology are the main reasons for this paradox. Clearly, this task is very complex and involves numerous feedback loops. The network plan is a specification of implementation details for the network configuration in order to provide demanded services at the right time and in order to achieve maximum profitability.

Services demand forecast is the main driver of planning. This forecast can reach a time horizon of 10 or 20 years. The forecast for immediate services demand generally has a sufficient degree of certainty, but sometimes it also comes in quite precise details. These certainties and details decrease for a longer forecast. A network plan provides services demand over the complete planning time horizon but it should deal well with the *non-linear* nature of the planning problem. In practical situations, there is not enough data to forecast the future demand and there is a big chance that new services may emerge or existing services may disappear. Therefore this thesis uses a combination of the numerical methods of operations research (OR) with the heuristic search techniques of artificial intelligence to replace traditional linear methods of broadband network planning.

2.5 Metropolitan area network planning

A metropolitan area network (MAN) is a network connecting many LANs located in different locations. It usually covers a metropolitan city or has a geographical scope range from 10 km to a few hundred km in length. The network carries several types of traffic simultaneously including data, voice and video traffic. It provides asynchronous, synchronous and isochronous services. Figure 2.3 illustrates a metropolitan area network.

Given existing infrastructure data as well as customer information such as the number and composition of customers and demanded types of services, the planning problem is to determine the best network technology to be deployed, bandwidth capacities for all the network links and the switch processing capacity.

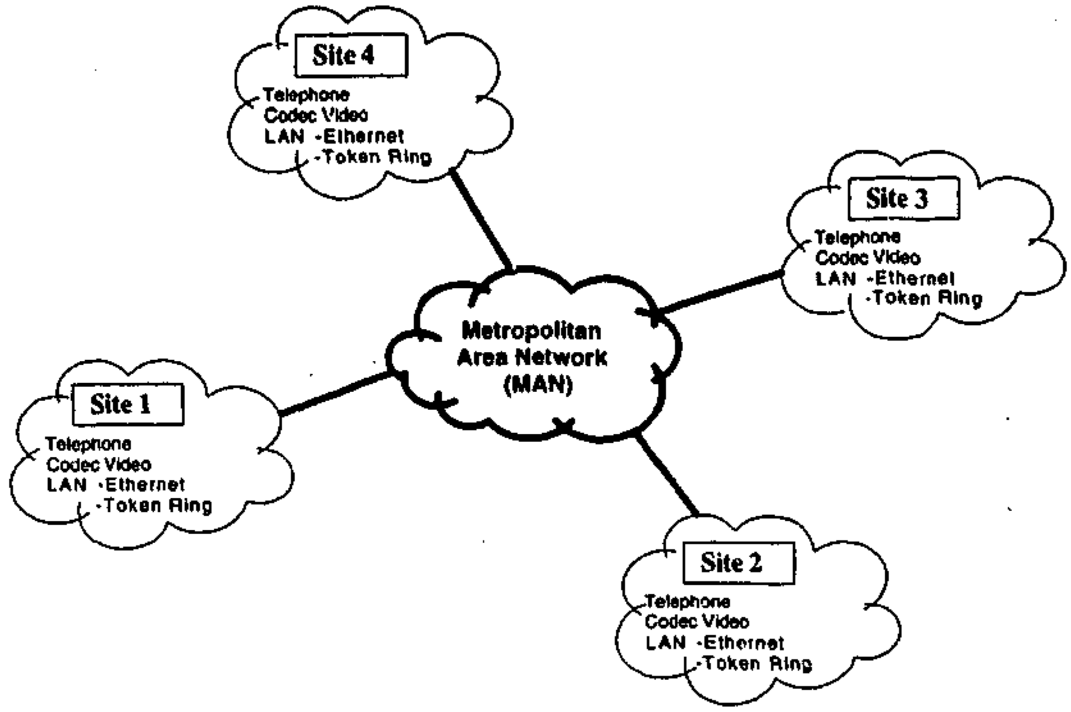


Figure 2.3: Metropolitan Area Network (MAN)

In this thesis, MAN planning process uses *network profitability* for selection of an optimal solution but considers all the factors previously mentioned. The quality of network plan depends on the correct assessment of many factors including:

- Types of services demanded by customers;
- Characteristics of all the demanded services;
- Traffic load resulting from provisioning of all services in terms of bandwidth;
- Information about the planned area, e.g. existing infrastructure and customer density;
- Possibility for enhancement of existing network in order to match the standards or range of services required.

A metropolitan area network continuously develops according to the advancement of support technology and demands for telecommunications services.

The following activities may happen along with the network development and should be considered for the planning process [14]:

- Network reconfiguration to anticipate demand increase;
- Network upgrade to match new standards and technology;
- Introduction of new services;
- Evolution towards ATM-based B-ISDN.

2.6 Intercampus network planning

Figure 2.4 shows an example of an intercampus network. The methods, criteria and issues of the MAN case also apply for the intercampus network. However, in this thesis, the planning is focused on the enhancement for the core network of the existing intercampus network. It does not include any change in the existing LAN. It assumes the LAN uses shared existing 10 Mbps UTP Ethernet LANs.

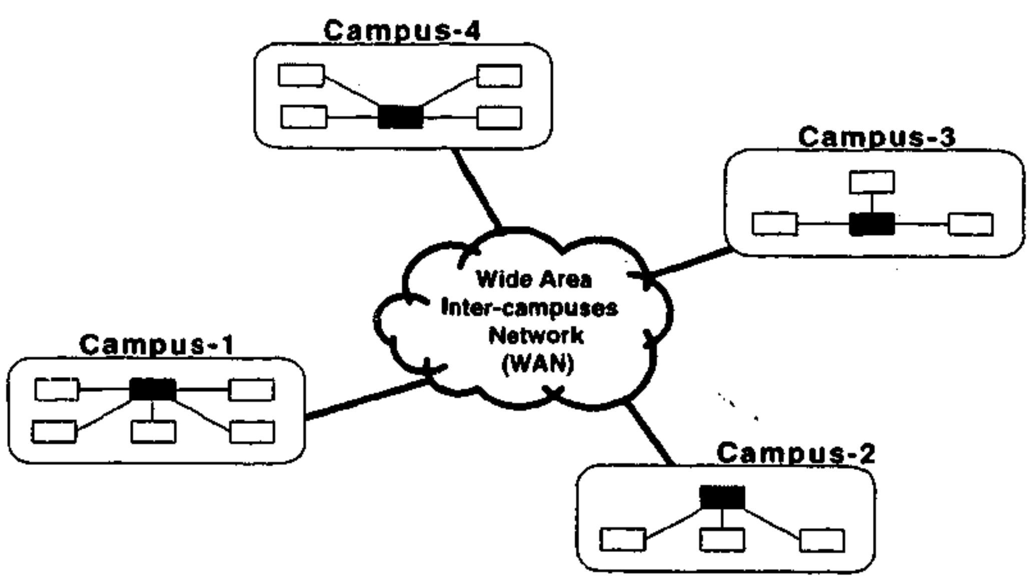


Figure 2.4: Intercampus network

Intercampus planning is required to suggest a network plan over a certain period of time (e.g. 5 years). It should maximise utilisation of the existing infrastructure and propose additional infrastructure where necessary. The most common objective of the intercampus network planning is to upgrade network capacity and performance to meet new and future needs. Of course, the integration of all communication facilities within a single network is also an important pursuit [15].

2.7 ISDN access network planning

ISDN is a network that provides end-to-end digital connections for users to attain access to voice, data and imaging services through a single facility rather than a multiplicity of networks [16]. Figure 2.5 illustrates an ISDN network.

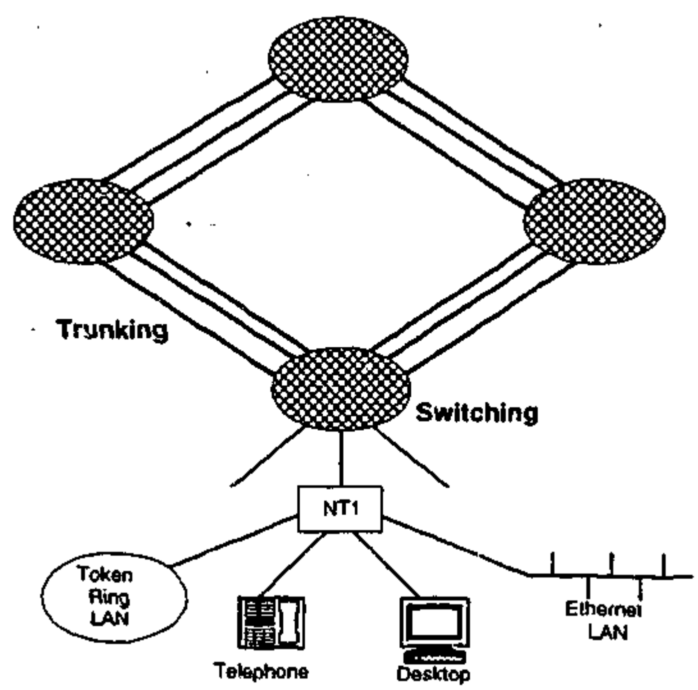


Figure 2.5: ISDN Network

This thesis focuses on the aspect of cost when upgrading the PSTN Customer Access Network to an ISDN access network. The planning problem is to determine the incremental cost for the upgrade. It determines the cost from the exchange to the customer premises and also includes upgrade costs, equipment costs, labour costs for installation and updating of the management systems. Planning should provide costs for a complete and partial rollout of ISDN. It should also provide an estimation of economic performance of the different technologies.

2.8 Software for automated planning of broadband networks

The availability of softwares which adopt a more comprehensive approach to broadband network planning is highly demanded. The ever increasing technical complexity of the problem and diversity of network services and technologies make the planning task too difficult if not impossible to be performed without a software assistance. The main concern is to how to make *planning knowledge* as well as network infrastructure and equipment data accessible in computer systems. An appropriate software would help increase the efficiency of the broadband network planning process [17].

However, it should be noted that a software would be unlikely to be used as a substitute for a skilled planner. The software aims only to assist the planners in some aspects of the planning process. Therefore, the software should be able to let human planners take control of aspects where they obviously have strong advantages.

2.9 Conclusion

This chapter has described the many inter-related factors involved in the broadband network planning process. It has shown an interplay amongst technical, economic, and many other factors in the lengthy process of creating a network plan. It has also briefly discussed the technologies available for the broadband networks.

The chapter has elaborated a planning method used in this thesis in terms of hierarchical planning layers. It has also clarified the optimality criteria used for making a final decision. This chapter has briefly described metropolitan area network (MAN), Intercampus network, and ISDN access network planning. Lastly, it has pointed out the necessity for an automated planning tool for broadband networks.

The next chapter, chapter 3, examines approaches and relevant techniques to solve the broadband network planning problems. It evaluates existing approaches on building planning softwares from literature and identifies remaining problems for further research. Chapter 3 also shows the need for the research reported in this thesis and describes the contributions which this thesis attempts to make.

Chapter 3

Approaches to Planning

The literature related to this work is very broad. Related subjects include broadband network planning applications, softwares for planning, *artificial intelligence* (particularly *constraint-based reasoning*) and *object-oriented* methodology. This chapter aims to set out a common understanding of the broadband network planning problem through selective references to the available literature. It describes potential approaches to deal with the complexity of the tasks. The chapter also discusses the remaining problems of existing approaches and points out research focus for this thesis.

3.1 Broadband network planning problems

In this thesis, the broadband network planning problem is viewed as a *constraint satisfaction problem* (CSP). Fundamentally, CSP is a problem constituted of a finite set of variables and a set of *constraints* that restricts the values the variables can concurrently take [18]. Each variable is related to a finite domain and the variable domain is a set of all possible values that can be assigned to the variable. They could be Boolean, numerical or symbolic variables.

The task is to allocate a value to each variable satisfying all the *constraints*. A CSP solution is then found from a choice set which does not violate a specified set of *constraints*. CSP also includes optimisation problems studied in the field of operations research [19].

Searching for a planning solution which satisfies all *constraints* is the hardest problem. In the first place, a constraint satisfaction problem solver should have the capability to generate some solutions which are all consistent with the *constraints*. The assignment of different values to the same variable frequently brings different solutions. Secondly, it should be able to produce one solution which is superior to the others.

This work aims to realise a software for dealing with the planning problem. The problem is to find an optimal solution where some application-specific criteria are used to determine the *optimality*. Planning knowledge are essential for finding an optimal solution without necessarily looking at all the solutions. This specific problem is also called a *constraint satisfaction optimisation problem* [18].

There are at least three classes of techniques which are related to CSP problem solving, namely *problem reduction*, *search*, and *solution generation*. Problem reduction is carried out in two ways, namely by eliminating redundant values from the domain and variables and by tightening the *constraints* (can be based on the domain-specific knowledge) so that fewer compound *labels* satisfy them. In CSP, a variable-value pair is called *label*. Problem reduction results in smaller domains of variables and tighter problem *constraints* so that there are fewer *labels* to consider.

The basic procedure is to select one variable and its value (*label*) at one time and to ensure a newly selected *label* is consistent with all the previously selected *labels*. If an assignment of current variable values fails, then an alternative value, when available, is selected. The problem is solved when all variable values are successfully assigned.

There are two beneficial characteristics of the CSP search space, namely finite search space and stable depth of the tree [20]. The efficiency of the search can be improved by pruning the search space that contains no solution. Pruning search space really means reducing the domain size. Therefore problem reduction can be done at any stage of the search.

Planning is a prevalent issue in the field of artificial intelligence and has been an active research topic since the early period of AI. *Planners* (planning systems) have been divided into two major classes [21]:

- *domain-independent planner;*
- *domain-dependent planner.*

A classic *planning system* reasons about the effects of actions and concentrates on identifying sequences of actions to meet certain goals when formulating a *plan*. A system which is built based on the classic approach falls into the domain-independent category. It uses a representation scheme and reasoning mechanisms that can be applied to a large diversity of problem domains.

This thesis is related to the domain-dependent planning systems which use *domain knowledge* to come up with an efficient search. Domain-specific *knowledge* plays a dominant role within such a planning system. Efficiency is achieved by encoding domain specific *knowledge* into the problem solver. Although this thesis is about a *domain-dependent planning system*, the proposed planning system architecture uses techniques and mechanisms which are generic in nature and can certainly be applied to other domains as well.

3.2. Existing approaches

Several papers have been published on computer-based tools for solving network planning problems [22-26]. In this section we review related works which focus on the use of different types of *knowledge* and methods within a single planning software. There have been also related computer-based tools published for network or equipment fault diagnosis [27-29]. The two types of applications, diagnosis and planning/design, actually apply the same principles when modeling real world entities. However, diagnosis uses models for analysis and planning/design uses them for synthesis purposes [30].

Mathematical models have been exercised to route traffic through networks so as to satisfy the demand forecast between all pairs of nodes. The given planning problem was characterised by capacity constraints associated with individual links. This problem is known as *multicommodity network flow problem* in the operations research literature [22]. The solution objective was to find a solution which minimises total routing cost. The model was used to construct a system of equations/inequalities solved through OR techniques. Mathematical modeling has a strong theoretical basis and high degree of precision so that it is very powerful in some particular cases.

Unfortunately, mathematical models have limitations on representing fairly large aspects of real world problems. Real world problems are too complex because they can have a very large number of variables and *constraints*. Mathematical modelling is frequently impractic ble for solving such problems [24].

The use of heuristics have been considered as an important technique that provides solutions which are not necessarily optimal but sufficiently good. It poses a strong potential to be applied to real world situations. Heuristics are highly suitable where the following conditions are applicable [31]:

- there is no efficient algorithm converging to the optimal solution;
- heuristics can speed up the process of finding solutions.

Developments in artificial intelligence (AI) techniques have used heuristics to deal with real complex situations. A technique known as *rule-based* (production) system has been effective for representing *knowledge* in a more flexible manner compared to the conventional mathematical techniques. AI applies *rule-based* systems to deal with telecommunications network planning problems. AI techniques have better performance on model and plan revisions compared to the methods of mathematical modeling [24].

Rule-based systems have been reported to be used for a network diagnostic system [32]. This work was exploratory research carried out to use *rule-based* system technology for diagnosing the AT&T switched network. The *rule-based* method was used for improving an heuristic approximation technique on dynamic analysis to solve rural network planning problems [24]. It has been successful to provide solutions to the problems which are impracticable to be modelled mathematically in a quite efficient manner.

However, *rule-based* systems also have some disadvantages. They are usually limited to discrete variables and do not usually provide optimal plans. It is difficult to include optimisation criteria in purely *rule-based* reasoning. On the other hand, mathematical models are not able to represent most aspects of real life problems, such as the *constraints* on finding solutions to network planning. Therefore an integrated approach that is based on both rules and mathematical models has also been adopted [27]. By doing so, advantages of both types of *knowledge* which are separately encoded in suitable ways have been exploited. Quantitative *knowledge* about networks can be better represented in mathematical formulas, whereas heuristic can be effectively expressed in symbolic rules.

A *model-based* approach aims to cope with the limitations of the *rule-based* approach. It is more robust because the *knowledge* used comes from a deeper understanding about the system rather than shallow *knowledge* of *rule-based* systems. However, *model-based* systems require highly intensive computation as the complexity and magnitude of the problem increases [28]. Therefore the use of *model-based* reasoning should be complemented by the use of other techniques that can increase the speed.

The integration of rule systems and database management system (DBMS) has also been explored [33]. Consistency between the data and *knowledge* base is the main concern here. This approach varies based on the decision about where to put the rules. Rules can reside in an application program which accesses the database. This results in the possibility that the database update is inconsistent with the knowledge base. Attempts have been made to cope with this problem. It has been done by coupling expert systems shells to database systems. But this method does not solve the consistency problem because the data and the rules are still separated in two different systems.

Realisation of a planning system that can improve its capability throughout its life is one important goal of AI researches. The system should learn from the previous cases it faced. The success of *case-based* methods to realise such a learning system has been reported in the literature [34, 35]. *Case-based* methods use similar cases from the past to find solutions for current problems. This method is highly suitable to a problem area where the domains are still poorly understood. Stored cases from the *case-based* approach could also be very efficient to solve particular network planning problems.

The use of the *case-based* method for network configuration and design has also been explored [26]. The problem of network configurations is characterised by an enormous range of possible choices at each stage of configuration and computational complexity has become the real issue. Cases are previous planning situations represented by attribute-value pairs comprising the problem specifications and the resulting plan description [36]. A simple learning mechanism such as *case-based* method can be very useful in practical situations. Adding new cases into a case-base is easier than other mechanisms of learning techniques. For example, obtaining learning examples for inductive learning systems or adding consistent rules are found to be more difficult than *case-based* learning [37].

This thesis aims to integrate necessary problem solving techniques which are essential to deal with the given planning problems into a single system. It seeks a solution through the combination of mathematical model, *model-based* approach and the use of rules to take advantage of each technique. In addition, the planning system attempts to benefit from the availability of *case-base* and database systems because these two modules can speed up the solution-finding process. This kind of integrated system is expected to results in a more powerful system by incorporating an enormous amount of *knowledge* required for the task of broadband network planning.

We exclude the connectionist techniques in our work because they are more suitable to deal with problems that require treating functions which are characteristically 'subconscious', such as visual perception and coordination, while our work is naturally suited to modeling functions where reasoning is seen to be very predominant [21].

3.3 Structuring the planning knowledge

Network planning requires significant amounts of diverse *knowledge* such as *knowledge* about network topology, switching nodes and internodal links, principles of telecommunication process, signalling protocols and many interacting *constraints* or parameters involved in network planning. *Knowledge* for cost optimisation is also essential in planning modern broadband networks.

There was already an attempt to use *frames* for representing the top-level appearance of customer network *knowledge* and rules for specific planning information. *Frames* describe objects and their attribute-values as well as the problem structure [38]. The *frame* has a number of slots used to define the various attributes of the object. The slots can have multiple facets for holding the attribute's value or default. The attributes can also have procedures associated with them which are executed when the attribute is asked for or updated. This effort was a good step towards the use of multi-paradigm *knowledge representation* for handling network planning problems.

In [25], *frames* were effectively used for representing a more static structure of networks such as nodes and links. This structure is independent of technology and services used and can be replicated easily through either copying or inheritance. Another kind of *knowledge* represented by *frames* was a design plan. This *knowledge* contains associated templates that form networks and event-sequence methods that will direct a process of configuring components. Nevertheless, it had both *frames* and rules implemented in Prolog which is based on chronological *backtracking*. This technique might be only appropriate for some limited applications and not suitable for dealing with the complex broadband network planning applications.

On the other hand, the *model-based* approach provides a clear and declarative paradigm to represent the enormous complexity of the network planning tasks. *Model-based* systems contain models of elementary network's components and all input-output parameters that construct inter-relationships amongst components [28].

There are at least two important elements related to the planning problem solving architecture, namely problem solving process and the *knowledge* used. A holistic approach to deal with network planning tasks suggests the need for an integrated problem solving architecture.

3.4 Constraint knowledge

Constraint satisfaction is a term covering a variety of methods of AI and some other related softwares. Freuder [19] suggests that a wide variety of methods can be brought to bear on *constraint satisfaction* problems. They include *backtracking, hill climbing, constraint propagation methods* and *operations research.* There are two main ingredients of these methods, namely search mechanism and the *knowledge* to guide the search. The initial idea of solving planning problems is to search for solutions exhaustively (*generate and test*). The *knowledge* is then used to guide the search toward the right direction or prevent the search toward the wrong one. This *knowledge* constrains the problem solver in performing the search.

Surveys on the algorithms for *constraint satisfaction* problems have been done by researchers [39, 40]. The algorithms for the CSP can be classified into three categories, namely *generate and test*; *backtracking* algorithms; and *consistency* algorithms.

The *generate and test* method searches exhaustively from all possible solutions. Each possible combination of the variables is systematically generated and tested to see if it satisfies all the *constraints*. The solution is the first combination that satisfies all the *constraints*. This method is a blind search method and does not take advantage of any *knowledge* to guide the search. It is prohibitive because it is a very inefficient method.

A basic technique for finding a solution of CSP is *backtracking search* [54]. The *backtracking* method searches a solution from the search tree structure in a depth-first manner and chronologically backtracks to the previous choice up the search tree when the search fails. This method has the danger of *thrashing* phenomenon. *Thrashing* is the phenomenon where the search unnecessarily keeps trying different alternatives repeatedly and unsuccessfully. This is possible because the search has not reached the culprit of the search failure when it backtracks.

There are some ways to prevent the *thrashing phenomenon*. An algorithm class of so-called intelligent *backtracking* identifies the culprit of the failure and backtracks to it directly when the failure is identified. *Dependency-directed backtracking* methods or *truth-maintenance* based systems apply this idea instead of exercising a pure chronological *backtracking*. However, the cost for implementing a truth maintenance system is also very expensive [41, 42].

Minton [43] proposed a simple heuristic approach to solving large-scale constraint satisfaction problems. His method is called *min-conflict* heuristic. The method starts with an inconsistent assignment for a set of variables and then searches through the space for possible repairs. The search is guided by a value-ordering heuristic that attempts to minimise the number of constraint violations after each step. The heuristic can be used with a variety of different search strategies. The construction of a good initial guess is a key problem for the repair methods. However, the *min-conflict* heuristic is suited to the problem with constraints which are all almost equally important. It will fail when facing problems with a few critical constraints. The *min-conflict* heuristic is also not suitable if repairing a constraint violation requires revising the current assignment.

The integration of both OR and AI techniques for solving *constraint* satisfaction problems has also been attempted [44]. This work mainly uses OR models for planning purposes. It aims to remove some limitations of both techniques and take advantage of their strengths. Various qualitative relationships were transformed into a form of mathematical relationship and symbolic models were integrated into mathematical models. This system has difficulties in formulating OR models and it also requires experienced OR modellers for various planning problems.

Consistency algorithm is a class of algorithms which attempts to ensure some degree of consistency in the *constraint* network of variable relationships. It aims to avoid *backtracking* and is complementary to the class of *backtracking* algorithms. The use of a constraint propagation mechanism inside a *backtracking* algorithm seems to be a very good compromise. Node and arc consistency are checked before the search. Node and arc consistency are considered very useful although it is not necessary to have all nodes and arcs consistent prior to the search [40].

This method can be coupled with the use of a heuristic based on structural *knowledge* such as variable ordering and pre-selected value instantiation [46]. Variable ordering is the way to achieve *constraint* satisfaction efficiently by choosing variables with fewest remaining alternatives first. This means the variables that participate in highest number of *constraints* are instantiated as soon as possible. *Constraint satisfaction* is most effectively achieved by choosing a value that maximises the number of options available for future assignments. It is done by implementing a mechanism to point out which value should be changed when modification is needed.

Constraints can also be ordered in an hierarchy. *Hard constraints* are the ones which can not be violated and soft *constraints* are the ones which are only preferences. The order of the *constraints* can also be indicated by using specific levels according to their level of importance [45]. The planning system considers only the "optimal" plan at any one time. Maximal satisfiability of *constraints* is determined by using certain metrics [46]. In practice, a cost metric is frequently used to determine the optimal plan. Unfortunately, it requires comparison of all possible solutions. However, the use of heuristic and planning *knowledge* can prevent the *deductive process* from having to examine all solutions.

3.5 Objects

The first critical step in order to achieve a satisfactory broadband network planning is to work out a systematic planning methodology. The preliminary suggestion to systematically approach planning problems is to model all planning system entities and relationships amongst important factors. The general conceptual representation of the planning problems is to model all entities which play a role in the planning activity [47].

The main advantage of using objects is to embed some forms of reasoning in the software system or to realise an automated deductive mechanism. A review on the use of *object-oriented* techniques in the telecommunications field has been reported [48]. In particular, the implementation of *constraints* using these techniques have been attempted [49, 50]. Objects can be used to represent and implement the constraint network of the *constraint satisfaction* planning. Objects are able to incorporate planning *knowledge* and objects can also be embedded into such *knowledge*. Therefore, objects can be seen as storage for working memory in a *knowledge-based system* [51].

The methods within the object have a perfect capability to incorporate the *knowledge* necessary for performing the planning tasks. Methods also include necessary task-oriented procedures to obtain a planning solution. This results in a dynamic *constraint network* where *knowledge* within an object is called on by other objects independently. The model is able to represent all planning entities. The *constraint propagation* mechanism which takes place within the *constraint network* is seen as a reasoning method.

However, Caseau [51] showed that a generic finite-domain *constraint* solver is not a panacea when it comes to solving large problems. This result suggests the need for a combination of the *deductive process* and heuristic knowledge.

There are three steps required for the use of objects and *constraints* as a reasoning mechanism, namely (i) construction of objects, (ii) description of relations between objects and (iii) description of object behaviour [52]. These can be achieved respectively by constructing object-based models of all planning entities, defining all binary relationships between objects and specifying *methods* which represent objects behaviour at the super-object level and objects of the lower level classes.

Most *knowledge-based* system applications are too complicated to be formulated in flat data structures [23]. *Object-oriented* programming simplifies both program maintenance and program modification i.e. it is easy to change and maintain objects. This would be a very powerful method for the maintaining the planning system.

3.6 Research focus

Most of the work in telecommunications network management has addressed problems of network monitoring and control, fault diagnosis and network maintenance. Planning is hardly addressed. This thesis particularly deals with broadband network planning problems. Broadband network planning is a hard problem which is tied to business planning and constrained by policy considerations concerning availability of capital, procurement of plant and manpower aspects. It is also heavily influenced by government regulations.

This thesis develops a model of a broadband network structure and function as the backbone of the software. The backbone contains models of elementary network entities and all input-output parameters that construct inter-relationships between the entities. By taking this approach, the enormous complexity of the broadband network and design model can be elegantly represented. The thesis also explores the use of stored cases to solve the planning problems more efficiently. Upgrading a set of past cases is relatively easy [36, 37].

Broadband network planning task comprises several subtasks with different problem features. Although *knowledge-based* systems seem to be the best solution for this problem, the task certainly requires algorithmic methods for some cases and also requires support from a large data-base. An integrated system is needed to assist planners in broadband network area planning. Table 3.1 shows how our proposed system compares to other works in the area.

Authors/ Developers	**General approach**	**Knowledge is incorporated in:**	**Problem solving technique**
Lee, Mehdi, Strand, Cox & Chen [22]	Operation Research	Data Base	Non-linear programming
Cappuro, Ravaglia & Giuli [53]	Procedural/ Algorithmic	Data Base	Planning methodology
Jennings [26]	Case-based Learning	Frames, Rules	Similarity matching
Costa, Climaco & Craveirinha [24]	Rule-based	Rules	Production system (Generate and Test)
Ferguson & Zlatin [25]	Rule-based	Frames, Rules	Constraint propagation technique
Proposed system	**Integrated approach**	**Objects**	**Object-based constraint propagation technique**

Table 3.1: The proposed system compared to the others

This thesis suggests the need to find an integrated problem solving architecture that enables the system to address each type of problem by a single integrated approach. The unified system should incorporate theoretical models of real world objects and processes, rules, relevant data bases and algorithms to find optimal or feasible solutions. The challenge lies in how to clearly define planning entities and to model each of them with all relationships between entities. This model is the core for the problem solving mechanism.

Having learned the advantages and disadvantages of available techniques it is very natural to seek a solution through the combination of mathematical model, *model-based* approach and the use of rules. This thesis proposes an integrated system which should result in a more powerful system.

3.6.1 Integrated planning system architecture

Many softwares described in the literature so far addressed only separate aspects or limited combination of tasks of the network planning process. A network configurer described in [25] provided only a set of solutions in terms of identifying a number of exchanges needed and service areas covered or distribution networks. Jennings [26] also described a way to assign the best link capacity of the network. The attempt of defining network traffic routing was described by [22]. However, none of these showed the system which is capable of solving such interrelated problems in an integrated manner.

This thesis describes research work for designing and implementing a software for network planners to deal with interacting aspects of network planning tasks. This thesis adopts an object-based model as the means to modeling the propagation of planning *constraints*. Effects in one part of the model would be propagated to other parts to cause further effects. Planning variables or parameters construct a system of object relationships to determine causal relationships amongst objects.

This thesis uses two main types of *knowledge*, namely structural and behavioural *knowledge*. Structural *knowledge* is the *knowledge* about planning entities in the domain and their physical configurations. Behavioural *knowledge* is the *knowledge* about functions of entities [30]. This thesis uses algorithmic or procedural programming techniques, heuristic rules and previously stored cases to make the system more robust.

Broadband network planning requires different kinds of *knowledge*. The differences also reflect the 'depth' and the way such *knowledge* is obtained and formulated. We propose an architecture for a computer-based planning tool that comprises multiple *knowledge* with a unified *knowledge representation*.

3.6.2 Incorporating different types of knowledge into objects

This thesis structures the broadband network planning problem solver as a set of cooperative objects which result in relatively self-contained processing modules. This approach is expected to result in a system that is easier to build, debug and maintain [50].

Having different types of *knowledge* work integratedly is a nice idea, but coordinating them to work together is very challenging. Weber et al. [55] placed the same pieces of shareable planning *knowledge* in each *knowledge-base*. Their sub-systems coordinate with each other through *message passing* instead of using shared memory. They developed a distributed forward chaining architecture and each subsystem has responsibility to send notification to other tools about modifications it has made.

In contrast to [55], this thesis proposes a constraint-based model at the centre of the system. In addition, it has different types of *abductive knowledge* to be integrated into a single system as the supports. The *constraint-based model* represents structural and functional aspects of the broadband network planning entities. The supporting *knowledge* can be that of a useful heuristic to solve planning problem, mathematical formulas or algorithmic types of *knowledge* and some stored solutions that may be useful to solve similar problems.

3.6.3 Case-base as means to preserve experiences

In order to incorporate learning ability, this thesis adopts a *case-based* approach into the system. Thereby, two objectives can be achieved: firstly, the ability of the system to quickly solve new planning cases based on its past experience; secondly, the ability to store the past solutions to improve system's *intelligence* without substantial reprogramming. These are the important features required of the planning software for this rapidly changing area of telecommunications.

The use of models, rules and mathematical formulas might not be sufficient to tackle the occurrence of potentially hard multi-criteria optimisation problems in the planning. The use of complete cases is a promising complementary method to the other techniques. More importantly, the use of a *case-based* technique in this thesis is expected to reduce the difficulty which usually occurs in the *knowledge* engineering process of the broadband network planning domain. The *Case-based* technique seems to be a more viable computational approach for the planning [56-59].

3.7 Conclusion

This chapter has set out a common understanding of the broadband network planning problem through selective references to the literature. It has described existing approaches to solve the problem, discussed the remaining problems and pointed out research areas of this thesis.

The chapter has shown several techniques which are useful to deal with the given problems. It has also shown how the newly proposed system compares with similar systems of other researchers/developers in solving the research problem. Lastly, this chapter has explained the focus of this research work.

The following chapter, chapter 4, describes a new approach for building a software for broadband network planning. It discusses the overall architecture of the proposed system in greater detail.

Chapter 4

Constraint-Based Planning System

Planning broadband networks is the process of finding an optimal solution given necessary input data as well as geographical area information, existing broadband services, and network infrastructures. The solution is the compound value of all the network planning variables. This chapter elaborates the overall architecture of the planning system and explains all components of the system.

4.1 Planning system architecture

This thesis proposes a *constraint-based model* for deducing planning solutions given a set of *constraints.* A computation mechanism is designed to deduce solutions using relationships between planning entities. The *constraint-based model* is the central part of the system architecture. However, computational complexity of a purely *deductive process* within a *constraint-based model* is a major problem. It carries a combinatorial explosion problem [60]. The overall architecture of the planning system is shown in Figure 4.1.

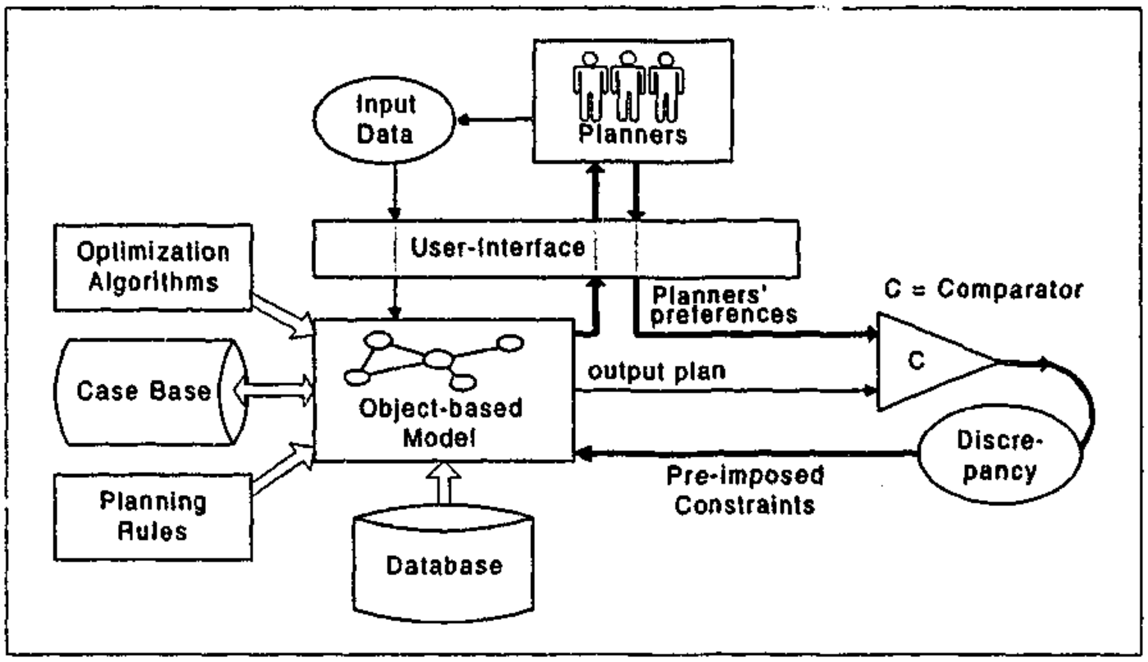


Figure 4.1: Planning system architecture

This thesis deals with the combinatorial explosion of possible solutions by using different kinds of *constraint knowledge*. *Constraints* are propagated throughout the planning entities via relationships between their variables. *Constraint knowledge* is used to guide the search and eliminate the unlikely alternatives from the whole search space. This strategy avoids unnecessary exhaustive searches within the model.

The *constraint-based model* is equipped with *abductive parts* to realise an effective system for solving network planning problems efficiently. The *abductive processes* take place to draw a planning solution by using *knowledge* from the *abductive parts*. The *abductive parts* include planning rules, solutions from past cases, and critical information from the database. The mathematically formulated method e.g. optimisation algorithm also plays its role as an abductive part because its results will be used by the system before the deductive process. Human planners 'guess' or select the plan by imposing their preferences and these preferences are also abductive part of the system.

4.1.1 Planning scenario

The system commences broadband network planning upon the completion of necessary input data or planned area specifications. These input data immediately become *constraints* for the planning. The system hence propagates these *constraints* over the planning problem structure. The *abductive process* takes place first and is followed by the *deductive process*. Because the *deductive model* covers all possible values, the system should eventually find a planning solution [29].

The planning system starts the process by building a complete structure of the planning *objects*. The structure is built according to the network types indicated by the initial input data. This structure is an *instantiation* of all *objects* involved in the planning process. *Objects* are initialised with certain default values and are connected with one another through the defined *messages* taking place between variables.

In the situation where there is no *abductive knowledge* applicable, the planning process is carried out 'blindly' using the deduction process to infer an *output plan* from the input data. When any inconsistency occurs, the *deductive process* is driven to find another solution. Discrepancies between output plan and planner's preferences also force the *deductive process* to take place again. This iteration forms the feedback cycle shown in **bold** in Figure 4.1.

The comparison between the planner's preferences and the output plan takes place within the planning process. Any discrepancy between the two triggers a binary state signal. The planner's preferences then become a requirement to be imposed as a top priority *constraint* in the current planning. The improvement is implemented by refining variable values (using the comparator in Figure 4.1) and again applying the *constraint propagation*.

It is important to note that only one planning solution is considered at a time. However, the planning system is designed to be capable of generating more than one solution which meets the requirements (input data) and the *constraint knowledge*.

4.1.2 Gaining efficiency

Computational efficiency is one of the main obstacles in *constraint-based* systems. The *abductive parts*, which consist of different modules of planning *knowledge*, are designed to make the planning process more efficient when integrated with the *constraint-based model*.

The software developed as part of this thesis also gains computational advantages through the way it implements the proposed architecture. The planning engine is implemented by defining planning entities as *objects* with *object-oriented modeling* in order to take advantages of the powerful technology from the software engineering and *database* community [33]. This thesis uses an *object-oriented programming* language as the basis to implement *objects* and rules which describe relationships between *objects*. Therefore, *objects* can be seen as storage for the working memory in a *rule-based* system [61].

This thesis uses the *object class* to represent the real world entity by grouping all data attributes and procedural operations into an encapsulated package. An *object* contains both data and procedures as a complete programming system. This thesis uses *objects* for modeling the application domain as well as planning rules for representing necessary decision making applied to the domain. Rules are used for both specifying *object* structure and behaviour and inferring new data from existing data. They are activated through a *message passing* mechanism.

4.2 Constraint-based planning

This thesis models the broadband network planning problem in the form of a *constraint network*. Figure 4.2 illustrates a *constraint network* of top-level planning *objects*.

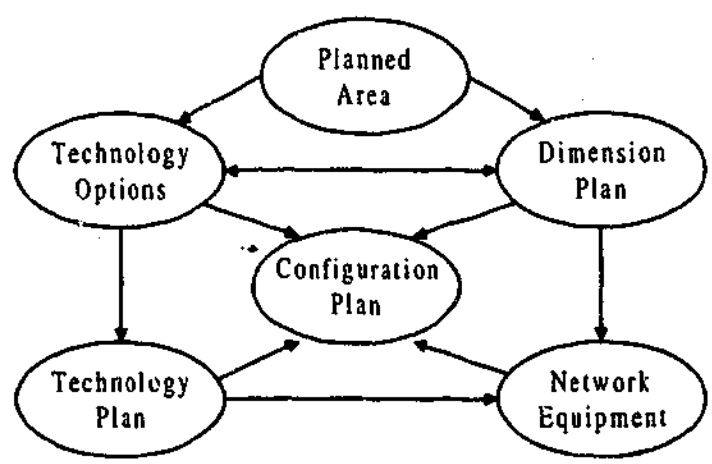


Figure 4.2: Constraint network

The *objects*, shown as ellipses, in the Figure 4.2 are composite *objects* which contain smaller *objects*. 'Planned area' consists of smaller objects whose attributes contain all necessary information for the planning process such as geographical area information, installed network infrastructure and existing services. Other objects apply the same principle. Each of them encapsulates smaller objects with associated attributes.

A planning solution is constrained by the many factors from the overall planning objectives. In Figure 4.2, an arrow indicates the existence of constraint that should be considered when assigning some attribute values of an object in relation to another object. This thesis applies a simple *constraint satisfaction* philosophy whereby *objects* must satisfy mutual *constraints* from their relationships.

In this thesis, *constraints* have the broadest meaning which include all *knowledge* that can be expressed in the form of mathematical relations, rules of thumb, specific optimisation methods or procedural algorithms. A *constraint* may be based on some planning rules, specific optimisation methods, or database retrieval results. It can take the form of a simple *argument* being passed between *objects*. Obviously all *constraints* are based on structural and domain specific *knowledge*. *Constraints* are actually the way to represent the problem characteristics.

The *constraint-based model* represents the planning problem structure through *objects* and *constraints*. *Constraints* represent the relationships between variables within an *object* and relationships between variables from different *objects*. Ideally, a solution for the network plan is a compound value of all the variables that does not violate a specified set of *constraints*. As the number of variables involved increases, the complexity of the whole network planning problem increases exponentially [62]. *Constraints*, in the view of the computation process, are used to reduce the complexity of the solution-finding.

Figure 4.3 illustrates a *constraint network* that describes more detailed relationships among some object attributes (or planning variables). Object attributes are represented as small filled handles attached to relevant objects.

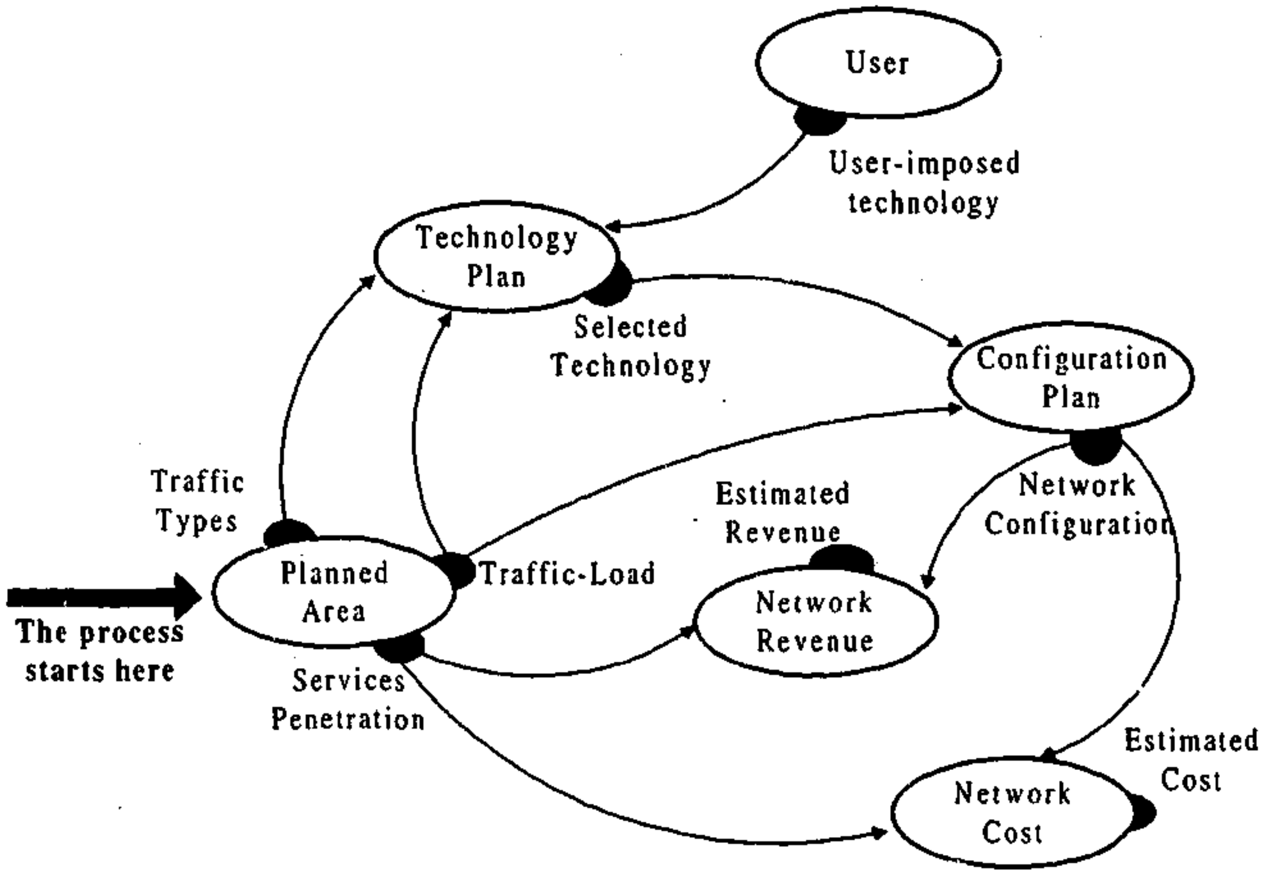


Figure 4.3: More detailed constraint network

Some planning variable values come from input data such as 'traffic types', 'traffic load' and 'service penetration'. These *constraints* are propagated to the 'technology plan' *object* and together with other *constraints* will determine the 'selected technology'. The 'network configuration' and its related 'estimated cost' and 'revenue' are also deduced through the *constraint propagation* process. This system selects an optimal solution out of all possible solutions which satisfy all the *constraints*.

4.3 Processing the constraint network

The most widely adopted technique to process *constraint networks* is *backtrack search* [40]. This technique systematically explores a search tree of all possible solutions. It constructs a complete solution by continually extending a partial solution and tests whether the current solution still has a chance to satisfy all the constraints. Figure 4.4 illustrates the backtrack search procedure for the broadband network planning problem structure.

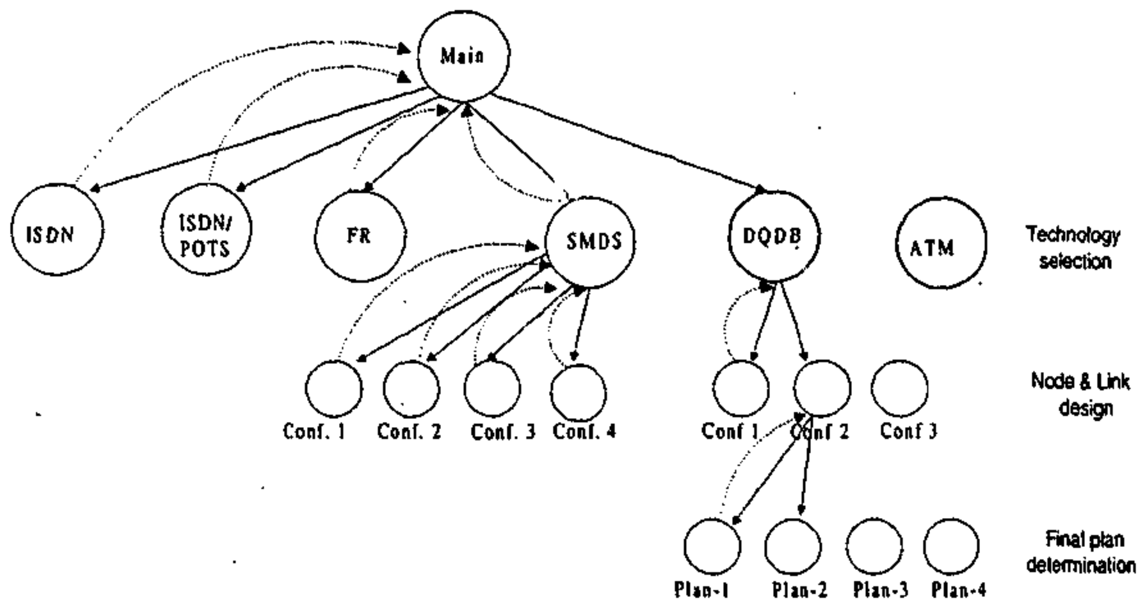


Figure 4.4: Backtrack search

The planning system searches for a network technology which potentially satisfies the all requirements as shown by the solid line arrows. The system backtracks (shown by dotted line arrows) whenever a certain technology does not possibly satisfy all the constraints. Let us assume that SMDS is chosen as the network technology to be deployed. The system then checks whether there is any possible configuration, based on the SMDS, which suits the planning requirements. If no configuration meets the requirements, the system backtracks until it finds a satisfactory solution. The complete solution should at last meet the financial constraints which determine the final solution.

Backtracking has a major disadvantage due to its potential inefficiency. Subtrees can be explored repeatedly while exploring different solutions which differ only in inessential features. This can happen when different values of a variable are assigned whereby the variable value does not affect the current failure. This phenomenon is called *thrashing behaviour* [63]. *Thrashing* can be avoided by detecting the real cause for search failure and backtracking directly back to the cause. However, the overhead to apply such capability is still considered too large so that it is still inefficient for real world problems [40].

This thesis uses a topological structure of the *constraint network* to guide the search and takes advantage of the planning *constraint knowledge* to preprocess the *solution space* before applying a *backtracking* algorithm. The planning engine uses a *deductive model* to represent the structure of the *constraint network* and applies *constraint knowledge* to preprocess the solution space.

4.4 Deductive model for constraint propagation

The *deductive model* describes all the *constraints* propagated from each broadband network planning entities to the other relevant entities. This approach makes modeling of the complex problem very simple. Figure 4.5 shows *constraints* propagated from some technology options for a network planning case. It shows the consequences of selecting the mixed ISDN/POTS, pure ISDN, SMDS and ATM as an access network technology to be deployed. This model should completely describe all *constraints* which are functionally necessary for the planning process.

ISDN/POTS:
- ⇒ assume 2 Mbps maximum capacity
- ⇒ assume voice traffic only

ISDN:
- ⇒ assume 2 Mbps maximum capacity
- ⇒ assume voice and data traffic only

SMDS:
- ⇒ assume 1.5 Mbps minimum capacity
- ⇒ assume 45 Mbps maximum capacity
- ⇒ assume data traffic only

ATM:
- ⇒ operates at the rate of 155 Mbps typical
- ⇒ support voice, data, video applications

Figure 4.5: Deductive model of access network technology options

Figure 4.6 shows the *deductive model* for the backbone network technology options. The use of SMDS as the selected technology restricts the interworking capability because SMDS can only interwork with the DQDB network. On the other hand ATM technology can interwork with other networks which are based on Frame Relay, SMDS or DQDB.

SMDS:
- ⇒ interworks with DQDB

ATM:
- ⇒ interworks with FR
- ⇒ interworks with SMDS
- ⇒ interworks with DQDB

Figure 4.6: Deductive model of backbone network technology options

The process of planning over the *constraint network* commences immediately given the input data. *Objects* which represent planning entities check whether there is any conflict between input data and its internal *constraints*. There is a predefined order for activation of the planning *objects*. Figure 4.7 shows the propagation of *constraints* over the *constraint network* using the *deductive model.*

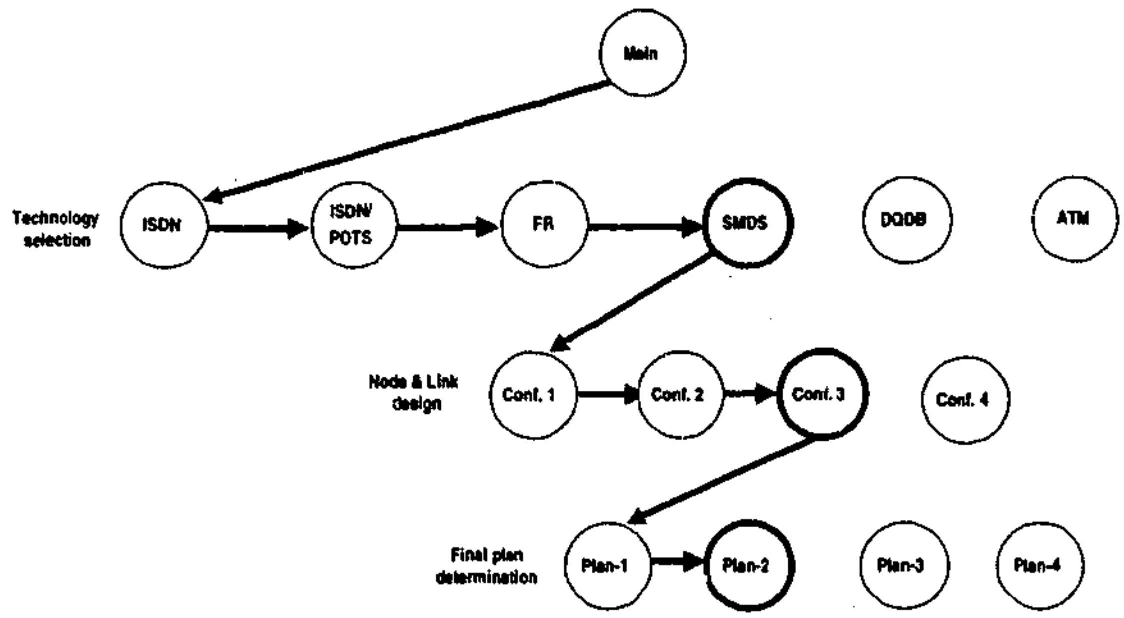


Figure 4.7: Constraint propagation using the deductive model

In this example, *ISDN object* is activated first for the technology selection of the access network. If any inconsistency occurs between all the input data and its internal *constraints*, it will send a *message* to trigger the next *object* at the same level to check the consistency. When an *object* can accommodate all the *constraints* and the input data, it will be selected as the current solution and will propagate *constraints* to the first *object* in order at the next level.

As illustrated in Figure 4.7, SMDS is eventually selected as the network technology to be deployed and propagates *constraints* to the relevant *objects* for determination of the network configuration. Dimensioning the network is the process to find types, quantity, and capacity of the network elements. Types, quantity and capacity of the nodes and links are the most important variables of the network configuration. Economy factors are final factors to determine the *optimality* of the solutions. The third level of the search tree in Figure 4.7 involves the *constraints* about the estimated cost, revenue and hence *profitability* of each solution.

4.5 Planning rules

The most popular system used to find a solution of complex problems like the planning problems is *knowledge-based systems* [64-67]. *Knowledge-based systems* are computer applications in which *knowledge* specific to the problem domain is kept separate from the procedure that manipulates it. This *knowledge* is usually coded in terms of declarative rules [68].

This thesis uses planning rules complementarily with the *deductive model.* The rules set represents the complexity of the problem in a very simple declarative way and it actually has a procedural interpretation as well. The rules set is very modular so that it can be added or omitted easily. While the *deductive model* contains 'deep' knowledge, the rules set contains 'shallow' knowledge about the domain. The combination of the 'deep' and 'shallow' knowledge is highly suitable for coping with the semi-structured nature of the problem and also for drawing conclusions with non-deterministic behaviour.

In the broadband network planning domain, the 'deep' knowledge is typically based on the planning specifications provided by equipment manufacturers, vendors and/or standardisation authority. The planning rules are typically provided by the experts (experienced planners) and become the *constraints* for searching subtrees in Figure 4.4. Figure 4.8 shows examples from the rules set containing relevant *knowledge* for the access network technology selection. Figure 4.9 shows examples from the rules set for technology selection of the intercampus networks.

Rule 1:
IF the traffic types include voice, data and video
THEN select the ATM technology
Rule 2:
IF the traffic types include voice and data; and
existing X25 network is reliable
THEN select the Frame Relay technology
Rule 3:
IF the traffic types include voice and data; and
traffic load > 2 Mbps
THEN select the SMDS technology

Figure 4.8: Rules for access network technology selection

Rule 1
IF any audio-video desktop application exists
THEN select ATM technology
Rule 2:
IF the network would have a combination of
PABX, CBR, and VBR streams
THEN select ATM technology

Figure 4.9: Rules for intercampus network technology selection

Planning rules play a crucial role for improving efficiency of the planning process. They speed up the process by guiding the search to directly select a solution avoiding unnecessary exploration of the subtrees which do not contribute to the desired solution. Figure 4.10 illustrates the search with the use of planning rules incorporated in each relevant *object*.

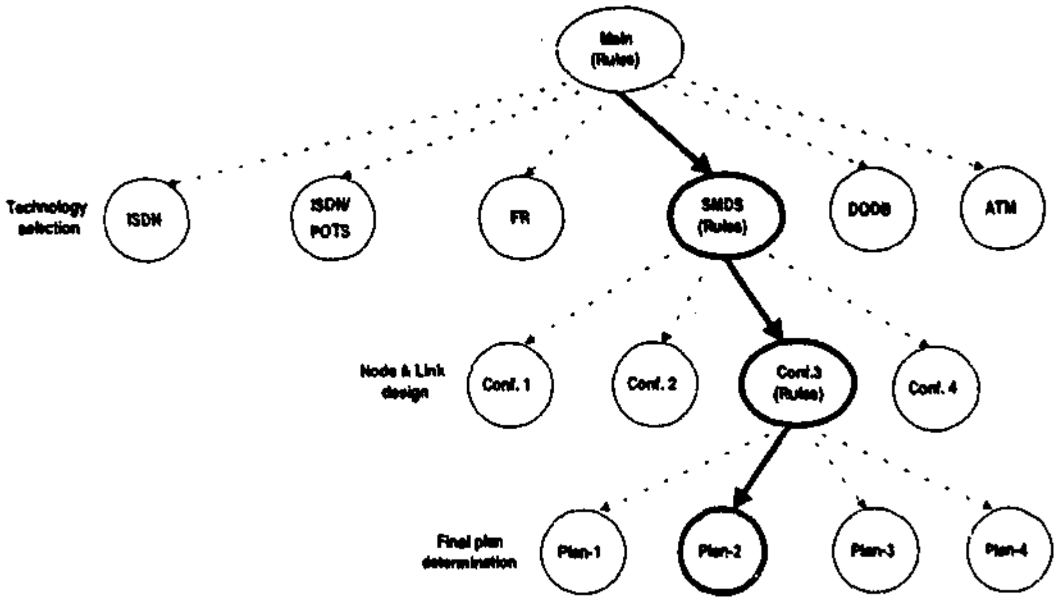


Figure 4.10: Guiding the search with rules

Planning rules directly select the SMDS technology at the technology selection process. This *abductive process* skips the *deductive process* that otherwise should take place over the network of objects in order to come up with a solution. The other parts of *abductive knowledge* (such as numerical optimisation result, similar solution from past cases and database retrieval result) also play their roles in directing the solution-finding in the same way that abductive rules do.

4.6 Backward propagation of current results

Forward propagation of *constraints* throughout the tree structure results in a solution which satisfies all *constraints*. This thesis implements the mechanism which enables the planning system to look at another solution or compare between different solutions. It requires a backward propagation of results in each step within the search tree. Figure 4.11 illustrates how current results, (lightly filled handles attached to relevant objects), are propagated backwardly in the *constraint network*.

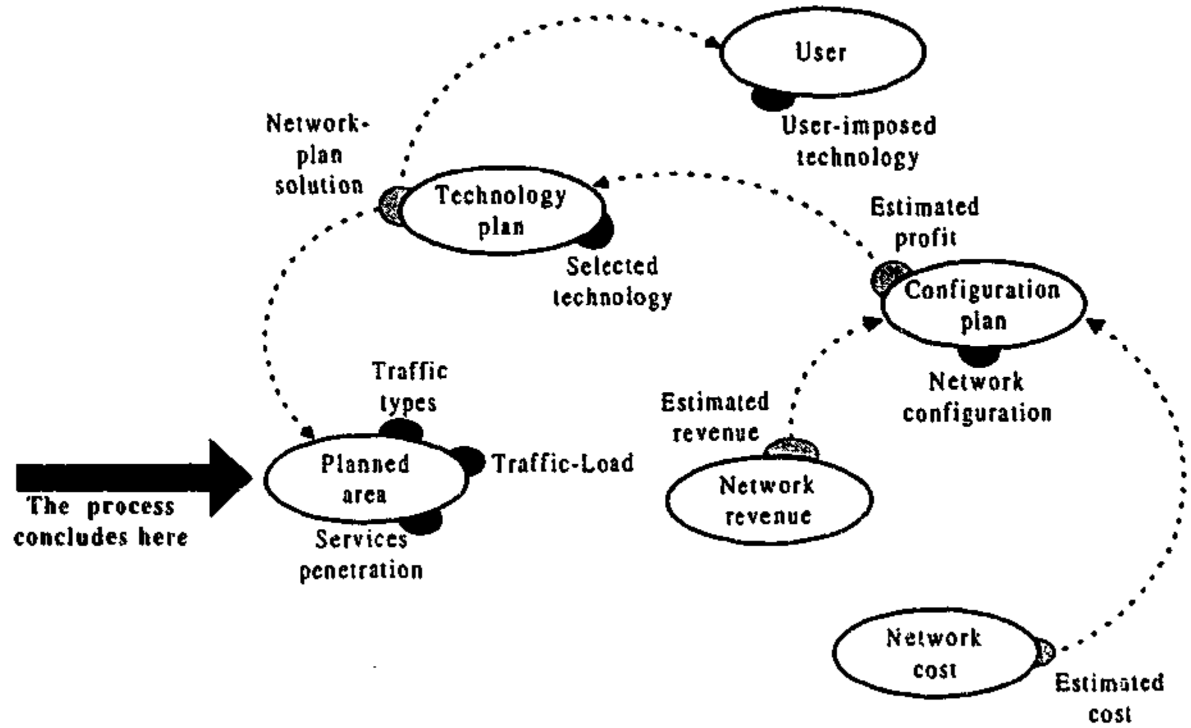


Figure 4.11: Backward propagation

Figure 4.11 shows a backward propagation of results over the constraint network for making evaluation or comparison of different solutions. Network cost and network revenue, for example, are final factors to determine the *optimality* of the final plans. These results are sent back to the configuration plan *object* for consideration when comparing different network configurations. The planning objective is to find a final plan which predicts the highest profit. The comparison between different technology options is also done in the same manner. The system instantiates different technologies and evaluates the results according to the predefined optimality criteria.

Figure 4.3 and Figure 4.11 illustrate how the constraint propagation mechanism takes place in two directions (forward and backward shown by solid and dotted arrows) over the network of objects.

4.7 Case-base module

The *case-base* module of the system aims to further improve the efficiency of the planning process. It provides heuristics for constructing a good initial 'guess' based on past experiences. Therefore, it can also be considered as a part of the *abductive knowledge*. The first proposal for a new plan is taken from the past solution to similar cases in terms of existing infrastructure, services to be provided and traffic load they have in common. This initial plan can then be refined to match all other *constraints*. The use of a case-base module improves the speed of plan initiation. This thesis uses a similar case, in terms of type of the area, traffic offered, services to be provided and geographical area conditions, to be used for the current plan.

The availability of the case-base module also improves the system's planning capability by expanding its case-base through its life, thus enhancing the quality of the proposed solutions.

4.8 Database support

Accurate data is essential for planning of a broadband network where network technology changes very quickly and too many alternatives are available to realise certain network specifications. Data not only helps the planning system to find realistic solution plans but also helps in improving the efficiency of the process.

4.9 Human planner role

The ability to look at alternative solutions and perform sensitivity analysis based on human planner preferences is very important in real life planning processes. This planning system includes the human planners as a part of the system components. Human planners can impose some variable values, and then the system would find another solution by taking the preferences as a high priority *constraint.*

4.10 Conclusion

This chapter has described the overall planning system architecture. It has discussed the *constraint-based* and the *deductive models* in detail as the central part of the system. It has also described how *abductive knowledge* helps improve the efficiency of the search.

The next chapter, chapter 5, discusses further all the pieces of *constraint knowledge* for directing the planning process. The chapter covers the topics of planning methodology, procedural algorithms, mathematical models, and rules of thumb. It also discusses *constraints* from the case-base and database modules.

Chapter 5

Constraint Knowledge

Finding a solution for the broadband network planning problem is like finding a solution for a puzzle problem. The solution finding features the involvement of different sorts of *constraint knowledge*. In this thesis, *constraint knowledge* consists of multiple types of *knowledge* that reduce possible solutions for broadband network planning. They reflect what should be held, what is preferred, what is restricted, and what is not allowed throughout the planning process. This chapter discusses different kinds of planning *knowledge* incorporated into the planning software.

Different types of *constraint knowledge* deal with different specific subproblems of the whole planning problem. In this thesis, they are all tightly incorporated into the *constraint-based model* to realise a true integrated system. They represent both relationships internal to the object as well as to the external *objects*. Basically each of them is the *knowledge* about one of the following categories:

- Broadband network traffic;
- Broadband network planning methodology;

- Broadband network technology;
- Network dimensioning;
- Economics of the network;
- Network equipment.

5.1 Input data as first constraints

All information regarding the planned area specifications are viewed as *constraints* for this planning process. As an illustration, the planned area may be divided into four zones, namely W, X, Y and Z as shown in an Figure 5.1. The locations of the exchanges are: A, B, C and D. The distances between A-B, B-C, C-D, and A-D are 4.7 km, 6.7 km, 8.3 km, and 7.3 km, respectively.

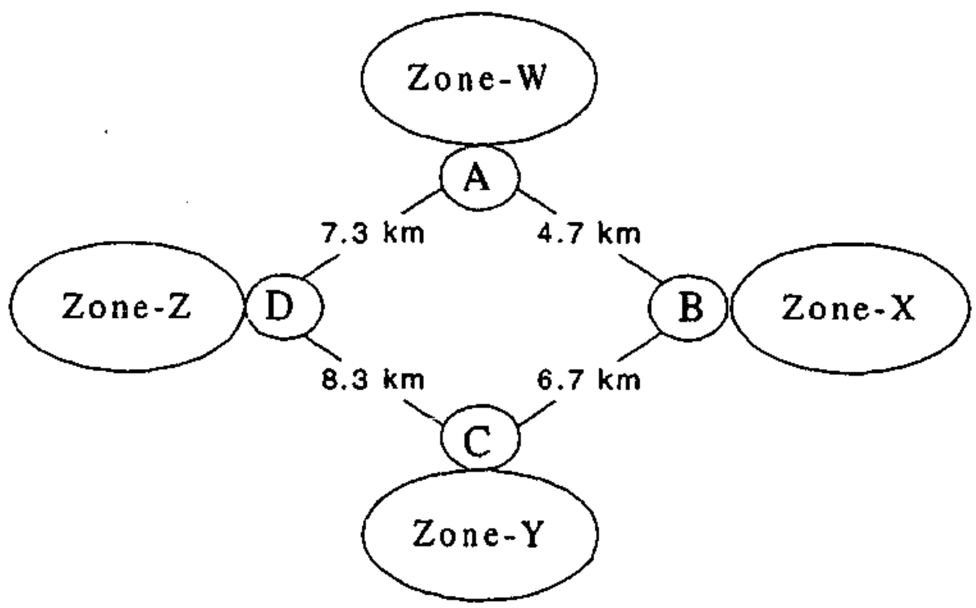


Figure 5.1: The planned area

The planning process should also consider the existing network infrastructure. Infrastructure of the existing backbone network have been assumed to be established as follows:

- Optical fibre links have been installed to interconnect the four exchanges as shown in Figure 5.1;
- The backbone area is an inter-exchange network that uses transmission systems of E3 PDH (34 Mbps);
- Each exchange is connected through point-to-point links to form a ring based on single-mode optical fibre cables.

The access network of the planned area is assumed to have the following specifications:

- Zone-W has been connected to exchange A, zone-X to B, zone-Y to C and zone-Z to D.
- All of these connections use twisted pair cables.
- Existing networks in zone-W, X, Y and Z are in a condition which allows the use of both High-bit-rate Digital Subscriber Line (HDSL) or Asymmetric Digital Subscriber Line (ADSL) technology without additional new cables being necessary.
- Existing cable systems in all zones are as shown in Figure 5.2:

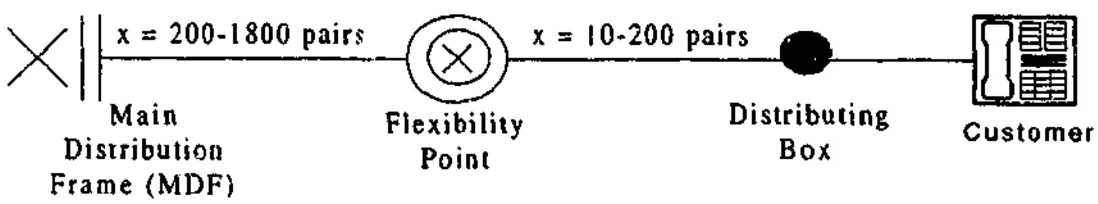


Figure 5.2: Existing cable systems

The planned area is assumed to have a Narrowband ISDN (N-ISDN) infrastructure as its existing network technology. The estimated traffic load from all nodes in a zone is feeded. The planning system should find an optimal solution by matching these *constraints* with multiple-sources of *knowledge* of the domain application.

5.2 Procedural algorithms

This software incorporates a systematic planning methodology that performs the planning tasks. The planning methodology is embedded within an algorithm which can be applied to generic environments and situations. An algorithm (or computer algorithm) is a sequence of steps designed for solving a specific problem. This section describes generic planning steps for broadband networks by using a certain planning methodology [53]. The section also gives an example of algorithm for determining a network topology.

5.2.1. Generic planning algorithm

This thesis carries out the broadband network planning through several steps as illustrated in Figure 5.3. This figure is an adaptation from a methodology described in [53] to this work with clearer steps and some modifications. The network planning steps include:

- Customer and services demand estimation;
- Traffic load analysis;
- Existing infrastructure analysis;
- Network planning, which consists of:
  - ⇒ Technology selection;
  - ⇒ Topology analysis;

⇒ Dimensioning.

• Economic and technical evaluation.

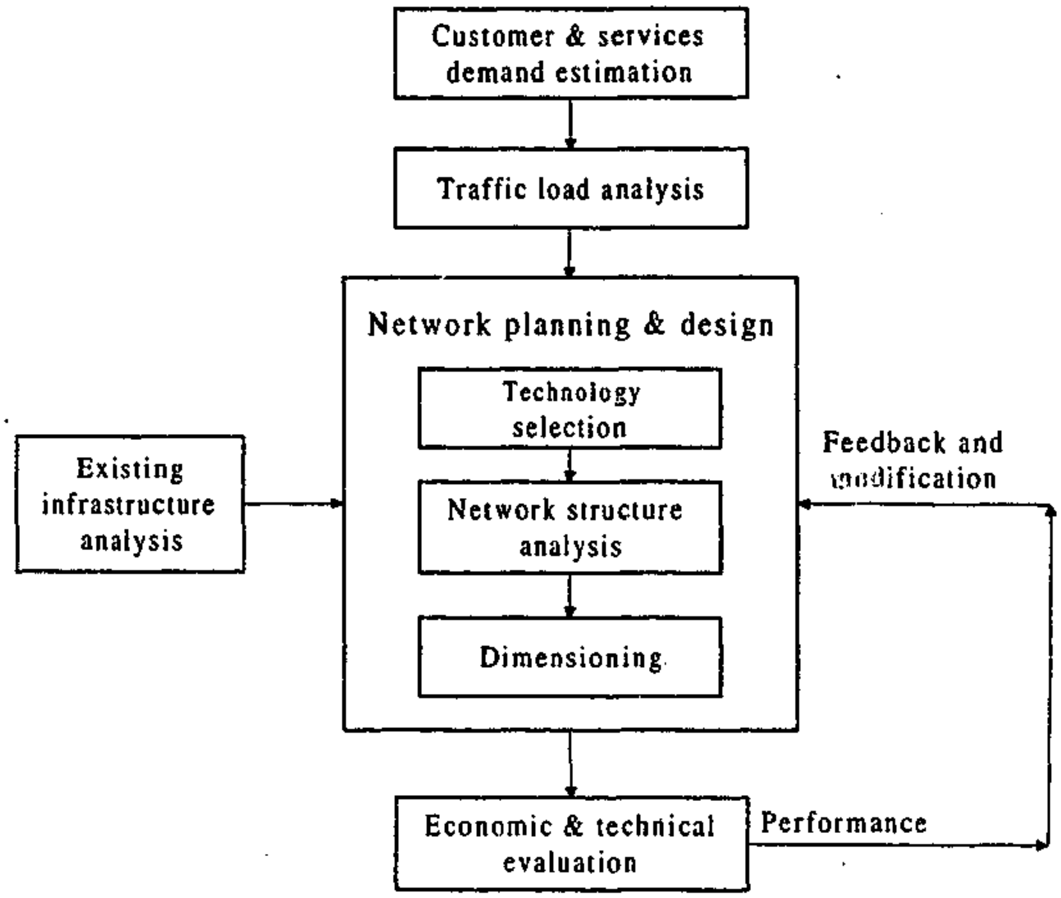


Figure 5.3: Network planning steps

Estimating customer and service demands

This step characterises customers in each area to be planned. It results in models of services and customers which will be used for forecasting demand. The models are very instrumental in the process of selecting all the services to be offered. The results from this process will be used for traffic load analysis.

This process identifies customers in terms of their needs for services and growth of these needs. For instance, business customers with a high number of employees most likely need services of videotelephone, telephone, high speed data transfer, etc. On the other hand, business customers with a low number of employees might only need telephone service. Customer group analysis results in a list of potential network services to be offered.

This step analyses each service based on an end-to-end performance objective and traffic attributes. End-to-end performance objective can be a Quality of Service, e.g. transit delay, and it would be used to determine feasibility of any public network to provide particular services. Traffic attributes are service parameters used to calculate traffic load resulting from all the services. Some important traffic attributes include:

- average bit-rate;
- maximum bit-rate;
- average peak-duration;

The objective of customer and services demand forecast is to select services to be offered in the planned area. The result of this process is a set of service candidates which still need to be evaluated in the later process in terms of their *profitability*. There are some factors to be considered in this process including:

- existing services;
- available network technology;
- new technology under development;
- investment plan.

Analysing the traffic load

Traffic load analysis uses the parameters of Penetration (P) of each service for the planned area, BHCA and Call Holding Time (CHT). The results of this step would give:

- Traffic load of different customer classifications;
- Total traffic load from different regions;
- Inter-regions traffic load.

This process determines the values of the above parameters for each service and customer group. These values are independent of any planned area but dependent on the service introduction phases. Table 5.1 shows an example description of service demands for a certain area at the introduction phase. The table reflects only one customer group demand at a certain period of time.

Service:	Penetration %	BHCA #	CHT
High Speed Data Transfer	2	7	-
Low Speed Data Transfer	30	3	-
High Speed Text Retrieval	20	8	-
Low Speed Text Retrieval	25	3	-
Video-telephone	10	8	360s
Telephone	90	65	180s

Table 5.1: Service demands

Taking the existing infrastructure into consideration

Broadband network planning is based on existing infrastructure which includes both access area and backbone area infrastructure. This process analyses existing network infrastructure such as current network technology, exchange locations, existing transmission lines and transmission system.

Selecting the network technology

This process selects the network technology structure which heavily determines the result of overall network planning. The introduction of a new technology enables the improvement of current services and creation of new services. This process considers all implications of the technology choices to the overall network plan. Planners should have an understanding about the technology developments and use *knowledge* about the existing networks and *constraints*.

Analysis the structure of the network

This process determines the structure of backbone and access networks. This process relies heavily on the experience of planners. The following factors should be considered in this step:

- existing infrastructure;
- traffic distribution;
- required access facilities for customers;
- functional characteristics of the network elements.

Dimensioning the network

This process determines network configuration which is based on the topology as well as required capacities. Network configuration is determined by the results of:

- network topology selection;
- required bandwidth;
- network element specifications and costs.

The optimality criterion used here is a maximum profitability based on potential revenues generated from the network reduced by all costs. However, it should also meet all technical requirements to satisfy the services demand.

This process determines network capacity for each topology selected. The network capacity is determined based on required bandwidth to provide all customer traffic. It includes the required bandwidth at the backbone and access networks. This step specifies types and capacities of network elements, types and quantity of transmission lines, buffer size, etc.

Evaluation of the economic and technical constraints

This process analyses economic performance of different feasible configurations. This step shows the network economic performance of the current plan and will use it as feedback for further optimising the solution. Feedback is very important for finding an optimal network configuration. It is based on economic and technical performance of the network because these two factors are the main risk factors of the network plan. The feedback process forms an iterative process for finding optimum network plan.

5.2.2 Specialised algorithm

The software assumes a given network topology when planning broadband networks. However, the capability of selecting the topology can be added into the software. *Minimum spanning tree* algorithm is a basic technique to find an optimum network topology [69]. This technique progressively interconnects several nodes within the network by establishing a graph representing the network. The assignment of link weights is based upon the distance between nodes, which is the prime component of the actual costs between links. One way to find a network structure with minimum total distances is to select the 'best' link possible at each stage of the algorithm. This specific kind of *minimum spanning tree* algorithm is called a *greedy algorithm* [69].

5.3 Mathematical models

Some aspects of the broadband planning tasks can only be tackled by the use of mathematical models [12]. The mathematical models vary in complexity. Some of them require simple calculations and some involve those which are very time consuming to be carried out in the routine planning process. Therefore they are incorporated into the planning software. This section describes some mathematically formulated pieces of planning *knowledge*.

5.3.1 Predicting the traffic from all customers

The planned area is divided into small-size zones for example 200m x 200 m areas. Some partitions can be grouped into a bigger area. The following parameters are predicted for each zone, namely:

- Number of customers in zone i (n_i) $\sum_i n_i = n$ ....(1)
- Number of customers in zone i from customer group k (n_{ik}). $\sum_i \sum_k n_{ik} = n$ ....(2)

The software can use the following simplification to avoid the evaluation of n_{ik} for each zone. It should be noted that this simplification is used for indicative study only.

- Define a customer composition model which is representative for each zone. For example, city area is the zone which has 85% residential and 15% business customers and rural area is the zone with 98% residential and 2% business customers. This can be set by the user at the beginning.
- Predict the number of customers in zone i (n_i) for each zone.
- Use n_i and the zone's composition model above for determining n_{ik}.

Each class of customers can be categorised further. Business customers are divided according to the number of their employees and residential customers are divided according to their income [70]:

LBC: Large Business Customer.

MBC: Medium Business Customer.

SBC: Small Business Customer.

HRC: High-income Residential Customers.

MRC: Medium-income Residential Customers.

LRC: Low-income Residential Customers.

Figure 5.4 illustrates a customer distribution for each zone of the planned area. The distribution of customers in the area Zone W, X and Y indicates that these are business areas. Zone Y has the highest density of business customers. Zone Z seems to be a strictly residential area where there is no business customer to be found.

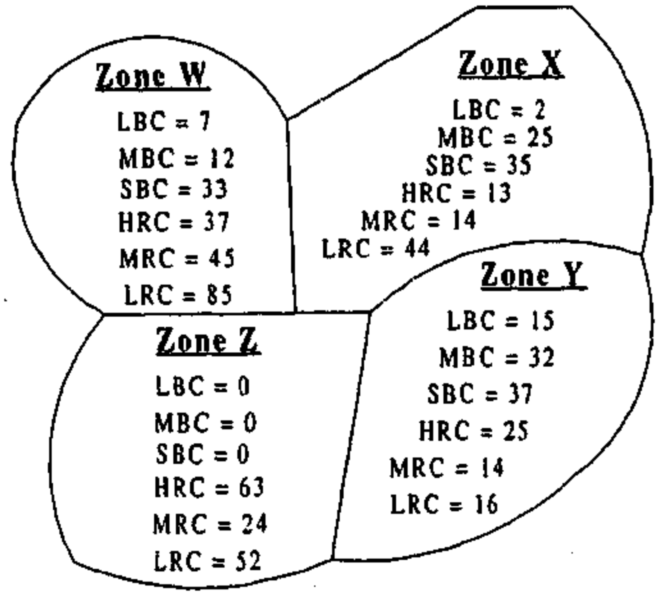


Figure 5.4: Planned area customers distribution

5.3.2 Estimating traffic from the network services

Network traffic can be categorised into voice continuous bit rate (CBR) and data variable bit rate (VBR) [71]. There are two important traffic parameters used for services with CBR namely, traffic volume (Erlang bit/sec) and BHCA (Busy Hour Call Attempts). On the other hand, VBR services use only BHCA parameter. The system converts all types of traffic into bits per second (bps). Table 5.2 gives some important examples of the broadband network services and their characteristics [53]. The table can be used to convert the traffic into bits per second.

Service Type	Max. Bit-rate	Ave. Bit-rate	Max. Duration
Telephone	64 kbps	38 kbps	180 s
Videophone	6000 kbps	2000 kbps	360 s
High Speed Data Transfer	10,000 kbps	1000 kbps	0.20 s
Low Speed Data Transfer	64 kbps	10 kbps	0.25 s
High Speed Text Retrieval	1000 kbps	1000 kbps	0. 30 s
Low Speed Text Retrieval	64 kbps	38 kbps	0.40 s

Table 5.2: Services characteristics

The planning system uses given data about service penetration and intensity for each customer class. The data include all types of services according to the area and the maturity of the service in the area. Table 5.3 shows an illustrative example of customer penetration data for the videophone service for a particular area at a certain period of time which reflects the service maturity in that area.

Customer Class:	Penetration (%)	Busy Hour Call Attempt
LBC	10	2
MBC	5	1
SBC	2	1
HRC	4	2
MRC	1	2
LIC	1	1

Table 5.3: Videophone service demand data

The estimation of total traffic offered to the network requires a large computational effort which can be a hindrance for the planners when there is no software available. The total traffic offered to the network is calculated by applying the following formula [1].

$$\text{Total traffic offered to the network} = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} T_{ijk}$$

$$T_{ijk} = (n_{ik} \cdot P_{kj}) \cdot (B_k \cdot BHCA_{kj} \cdot CHT_{kj}/3600) \cdot b_k$$

where: T_{ijk} = traffic volume in zone i, from customer class j and service k.

P_{kj} = Penetration of customer class j over the service k

$n_{ik} \cdot P_{kj}$ = number of customers in class j for service k within zone i.

B_k = maximum bit rate of service k

$BHCA_{ijk}$ = BHCA at zone i, from customer class j and service k.

CHT_{kj} = Call Holding Time or maximum duration of customers in class j for service k

$B_k \bullet BHCA_{kj} \bullet (CHT_{kj}/3600)$ = traffic volume from customer class j for service k.

$BHCA_{ijk} = (n_{ik} \bullet P_{kj}) \bullet BHCA_{jk}$

b_k = bidirectionality coefficient of the service k

The traffic estimation is performed for the indicative study in the association of a certain zone, customer class and service. It should be noted that B_k is bit-rate of service k while b_k is the bidirectionality coefficient of the service k. The value of b_k is 2 for services which require two-way traffic and 1 for the services with one way traffic.

5.3.3 Estimating the network profit

Network profit is the criterion used in network planning. It is represented by the optimal cash flows which involves calculation of net cash possibly generated by the network. This method examines cash flows which occur at *different points in time*. It evaluates a cash flow with C_1 at time t_1, C_2 at t_2, ..., C_n at t_n according to its *present value* [8]:

$$PW \equiv \sum_{i=1}^{n} C_i e^{-rt_i}$$

where r = discounting rate.

or, for continuos time: cash flows that are given as rate *c(t)* over time, where *c(t)dt* is the amount of cash flow in the interval *t* to *t+dt*.

$$PW \equiv \int_0^T c(t) e^{-rt} dt.$$

It is obvious that the cash flows calculation requires inclusion of all cost components and predicted revenue as well as other factors based on investment planning. All cost components such as equipment, installation, upgrade and management cost should be taken into account in order to have more precise cash flow results. However, the complete financial models will not be used in this thesis as the network *profitability* used for network plan evaluation and comparison is only indicative.

5.4 Rules of thumb

Planning of the broadband network need analytical *knowledge* which has strong theoretical foundations. However, the problem solving also frequently requires *knowledge* in the form of rules of thumb. This type of *knowledge* is an extract of different pieces of *knowledge* and is usually based on the experience of the expert in the domain application. Some rules for the network technology selection from an expert may look as follows [76]:

- Never select a less advanced technology than the existing one;
- Upgrade one step toward a more advanced technology if the traffic offered exceeds the maximum capacity of the existing technology.

The rules can even take a more strict form and need to be considered in order to reflect a realistic situation in a certain area being planned. Figure 5.5 exemplifies a pre-determined technology migration path that becomes a *constraint* when the system carries out planning tasks. In this example, a combination of PSTN and N-ISDN technology is selected if the network traffic load is not more than 2 Mbps. For the network traffic load that lies between 2 Mbps and 40 Mbps, SMDS is to be selected over a metropolitan area network. ATM-based technology should be selected if the network traffic exceeds 40 Mbps. These rules might be from a telco's strategic planning and management policy. They should be incorporated within a practical planning system.

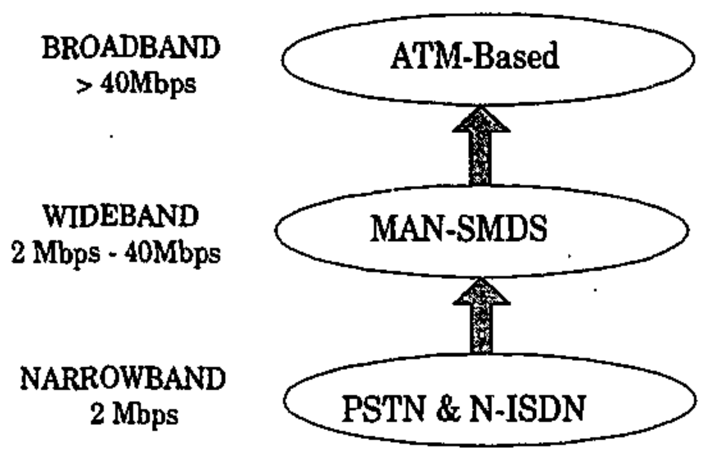


Figure 5.5: Technology migration rule

The following statements are other examples of rules for the intercampus core network technology selection.

- If any video desktop application exists then select ATM technology;
- If the network would have a combination of PABX, CBR, and VBR streams then select ATM technology;
- If network should have a high capacity (> 100 Mbps) then select ATM technology.

Network dimensioning determines the types, quantity and capacity of the network nodes, links, and buffer size. This process ensures that the selected network devices will provide a desired level of service. This thesis uses the combination of two methods to determine the types and size of network devices. The first method is based on planner's experience and intuition encapsulated in the form of rules. The second method is by using the *knowledge* of traffic data and application of mathematical formulae to traffic data as shown in the previous subsections.

In order to dimension networks effectively, *knowledge* about equipment specifications is very important. Planners are also forced to perform some sensitivity analysis to come up with optimised network dimensions. As an example, when the available bandwidth at any particular node is still lower than the customer bandwidth requirement in the service area, the following alternative solutions should be considered [12, 53]:

- to increase buffer size as long as it does not cause an unsatisfactory transit delay;
- to increase node bandwidth by considering the backbone network and the node processing capacity;
- to divide area into smaller zones with additional equipment.

Knowledge about network elements and expertise of the planners about the planning process and the features of the broadband network products play important roles for network planning. These kinds of *knowledge* are incorporated into the planning software as *soft constraints* and subjective rules.

As an example, *knowledge* about network elements and products plays important roles in planning networks. Network elements and products have specifications according to different vendors. A MAN element product, for instance, can function as EGW, CGW or SR depending on the equipment modules from which they are constructed. Each module also has technical specifications such as the number of interfaces available and transmission rate, etc.

5.5 Past cases make the solution-finding simpler

This thesis uses another source of *constraint knowledge* to speed up the search for a planning solution. It is done by retrieving a case-base where the planning cases from the past is accumulatively archived [72].

This system adopts a *heuristic* method to find the 'most similar case'. It selects the first case found to match adequately based some predefined case features. These features are selected from the case features which have same functional roles for the broadband network planning [77]. The type of the area being planned, total traffic offered to the network, and services to be provided in the area are used to evaluate the similarity between two cases. The system retrieves a similar case from the past and adopts it as the solution to start with.

Cases matching can also be done by using a numeric *evaluation function* or a mix between heuristic and numeric method. Numeric method is done by computing a score of the degree of match based on relative importance of the case features. The most similar case is the one who has a highest degree of match with the current case being dealt with.

5.6 Constraints from the database retrieval result

Database does not only help the planning system to find a realistic solution but also helps improve efficiency of the planning process [33]. Table 5.4 shows an illustrative example of a data piece from the *database* of the MAN switching system available in the market. The table includes information from the manufacturer such as capacity, type, and price.

Manufacturer:	AlphaComm
Call Capacity:	700
Line Capacity:	600
Load Capacity:	650
Type:	T123
Product Code:	SiX650
Year:	1993
Price($):	15,000

Table 5.4: MAN switching system data

The planning system not only needs the ability to automatically access the information as shown in Table 5.4, but the on-line access is also essential for selecting an *optimal* capacity of the network elements for a given traffic load. Price data, for example, is also required for deriving some indicative network costs which will be then used for determining network *profitability*.

5.7 Conclusion

This chapter has discussed all types of *constraint knowledge* for the planning software. It has elaborated the *knowledge* in terms of planning methodology, procedural algorithms, mathematical models, and rules of thumb. It has also explained how the cases from the past and pieces of information from the *database* become the *knowledge* used for restricting the number of possible solutions.

The next chapter, chapter 6, shows the implementation of all planning *knowledge* into a network of objects. The chapter shows how the *constraint knowledge* is incorporated into *objects* and how to propagate such *constraints* over the network of objects. Chapter 6 also elaborates the planning process within the software and discusses the results.

Chapter 6

Objects for Constraint Propagation

This thesis uses a *network of objects* to represent enormous complexity of broadband network planning. *Objects* model the planning entities, their attributes, functions and relationships amongst them. *Objects* use *message passing* as the mechanism for propagating the *constraints* over the relationships amongst them. The use of *objects* enables integration of different types of *knowledge* in a uniform manner. This chapter describes the structure and behaviour of the broadband network planning entities which are modelled as *objects*.

6.1 Objects

Objects facilitate the representation of both the planning entities as well as all the *constraint knowledge* [68]. They are instances of corresponding entities of the problem domain which can be computationally executed. An *object* encapsulates both data and processing capabilities in a single entity. It makes modeling of complex problems much simpler by focusing on each entity and its relationships with other entities involved in the planning. This thesis uses *objects* to implement a *computational model* which reflects the structure and behaviour of all the collaborating planning entities.

5.1.1 Attributes, functions and operations

An *object* is an abstraction of an entity which can be physical or logical. It captures only useful information and excludes features of the entity which are not relevant to the understanding of its functions. Figure 6.1 shows the structure of an *object*. The main elements of an *object* abstraction are attribute, function and operation [47].

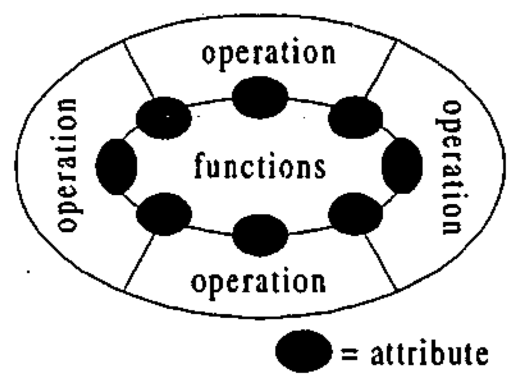


Figure 6.1: Object structure

The main characteristic of an *object* is the packaging of abstractions of the real world entities in an encapsulated description. Attributes represent the data in terms of *object* states or variables which can have values of a certain kind or range. Functions contain the procedure in terms of the *knowledge* maintained or the actions that can be performed by the *object*. Functions can be seen as the *object's* responsibilities to accomplish its goals as a part of the whole *objects* which collaborate one another.

An *object* collaborates with other *objects* to perform complex tasks through *message sending* [50]. *Objects* model the behaviour of the entities in terms of the operations they can perform. Operations specify which *message* the *object* can accept. The *object* states or variable values are hidden to the external environment. Operations act as the interfaces of an *object* to the external *objects*.

6.1.2 Object class

Object class is an abstract description which defines the common features of *objects*. A class is a generic specification for similar *objects* and can be considered as a template for a specific kind of *object*. An *object* actually is an instance of a class to which it belongs. Instantiation is the creation of a particular *object* from a description of its class.

Figure 6.2 shows a 'network plan' *object class* with attributes and operations listed in the middle and lower parts of the class box [47].

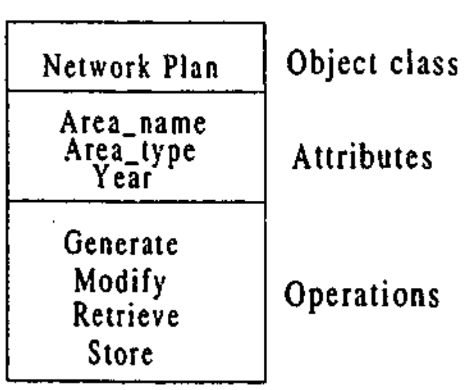


Figure 6.2: Network plan object class

'Area name', 'area types' and 'year' are attributes of the network plan. The network plan is generated for a specific time horizon (e.g. year). Valid operations for the network plan include: generating a plan, modifying a current plan, retrieving past cases (that consist of previous plans) and storing a satisfactory plan into a *case-base*. An internal code for the network class is shown in Figure 6.3 [73]. The first line indicates the *object* name and the third, fourth and fifth lines show the *object* attributes. The valid operations of the *object* are in the last part of the code. The detailed execution of the operations are implemented within methods. An *object* method may contain the implementation detail of an operation or any required function related to the *object*.

```
@interface NetworkPlan:Object
{
char      *area_name;
char      *area_type;
int       year;
}
- generate;
- modify;
- retrieve;
- store;
@end
```

Figure 6.3: Network plan object internal code

6.1.3 Inheritance hierarchy

Inheritance is a principle of structuring an *object class* onto other *object classes*. Inheritance expresses the semantics of extension, incremental refinement or specialisation [52]. Figure 6.4 shows a 'planned area' class which has two subclasses, namely 'access area' and 'internodal area'. Each of the latter *objects* is a specialisation of the 'planned' area' *object*. Both of them inherit the features of the 'planned area' as their superclass.

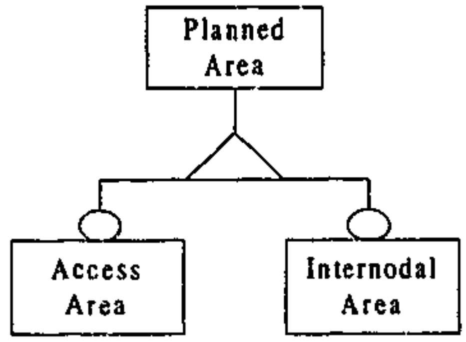


Figure 6.4: Planned area subclasses

Figure 6.5 shows a service *object* with a hierarchical structure based on the specialisation. Vidoephone, telephone, high-speed data transfer, low-speed data transfer, high-speed text retrieval and low-speed text-retrieval services all have some common attributes. Their common attributes are average bit-rate, maximum bit-rate and directionality. Each of them may have some unique attributes in addition to the common ones.

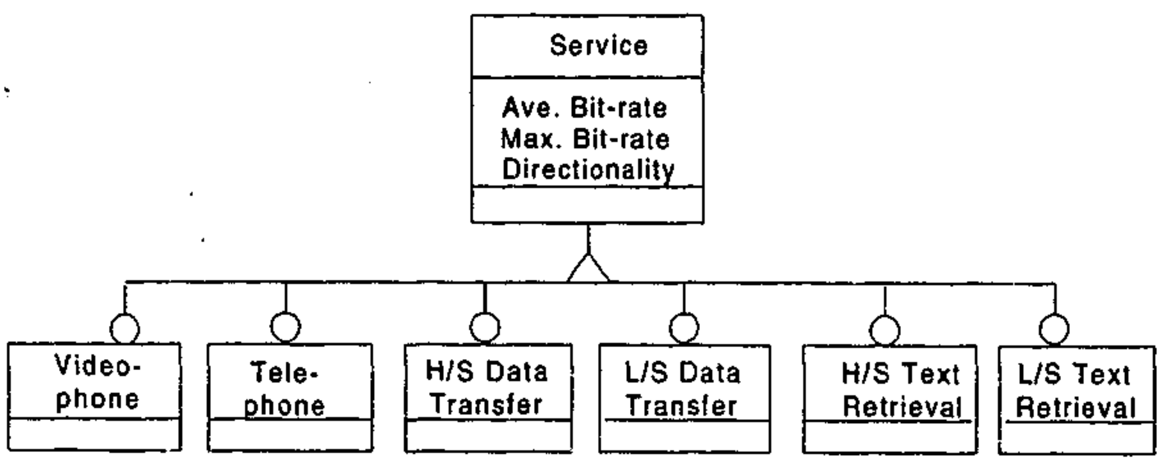


Figure 6.5: Inheritance hierarchy of 'service' object.

Figure 6.6 shows how attributes of the 'service' *object* are inherited by its subclasses. 'Service' *object* is a superclass for 'HSTextRetrieval' and 'Videophone' *objects*. Each subclass has its own specialised attributes and also adopts all of those of its superclass.

```
@interface Service:Object
{
        int     aveBitRate;
        int     maxBitRate;
        int     directionality;
}

@interface HSTextRetrieval:Service
{
        float   avePeakDuration;
}

@interface Videophone:Service
{
        int     callHoldTime;
}
```

Figure 6.6: 'Service' object and its inheritance objects

An *object class* can also be decomposed into a hierarchy of component *objects*. A component *object* is an instance of an *aggregate* class. Aggregation is an identification of part-whole relationships between *objects*. An example of the network plan class hierarchy which is based on aggregation can be seen in Figure 6.7.

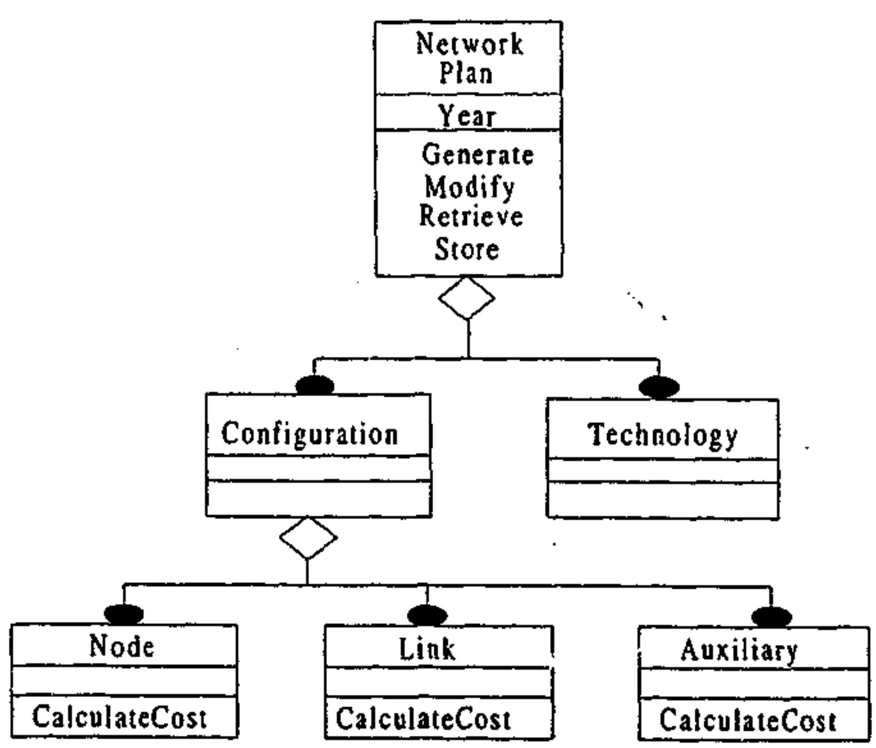


Figure 6.7: Hierarchical structure of 'network plan' object

Network plan consists of a complete specification of the network technology and all devices required at the network nodes and links and auxiliary devices. The network plan is an aggregation of both technology plan and configuration plan. Furthermore, configuration plan is an aggregation of the plan for network nodes, links and auxiliaries.

6.1.4 Relationships between objects

This thesis models relationships between *objects* that reflect which *objects* should collaborate to accomplish the broadband network planning task. Figure 6.8 shows four planning *object classes,* namely planned area, network plan, technology options, services and equipment, and describes relationships among them.

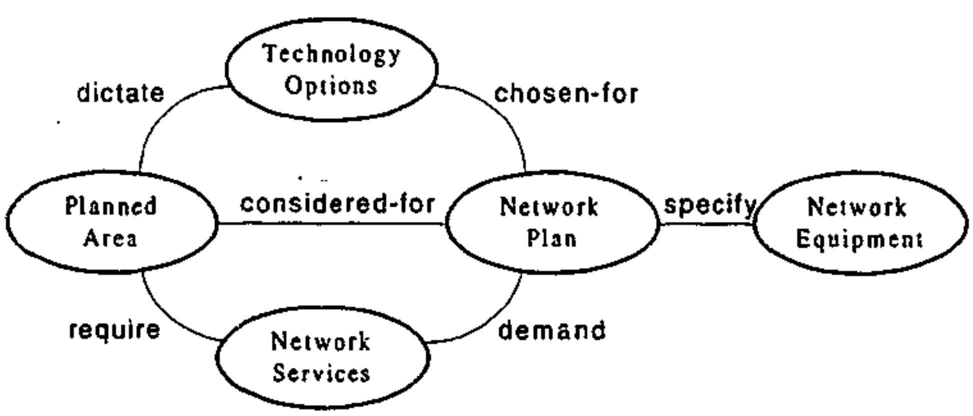


Figure 6.8: Object classes and relationships

Basically, a link between two classes represents a bidirectional relationship. However, link titles indicate only one particular direction and another direction is implicit in the figure. The following are the relationships between *object classes*:

- 'Planned area' specifications should be considered-for formulating a 'network plan'.

- 'Planned area' specifications dictate the options for the 'network technology options'.
- Telecommunications demand in 'planned area' require 'network services'
- 'Network services' to be provided demand 'a network plan'.
- An option from the available 'network technology' is chosen for a 'network plan'.
- 'Network plan' specifies all pieces of 'network equipment' required.

Planned area and the network plan *objects* establish links between themselves through *messages*. Figure 6.9 shows a method which functions as a link between the two *objects*. The software implementation for the system was written by the author of this thesis using the Objective-C programming language in a NexTSTEP 3.2 environment which is based on the Mach version of the Unix Operating System [73, 75].

This method sends a *message* containing 'traffic load' data from the 'planned area' to the 'technology plan' *objects*. The *message* will be sent whenever the value of 'traffic load' is set or changed.

```
@implementation PlannedArea
// A method implemented within 'planned area' object
- setTrafficLoad:(int)theTrafficLoad
{
if (theTrafficLoad) {
        trafficLoad = theTrafficLoad;
   }

        // Send 'traffic load' message to the 'technology plan' object
        [tPlanObj areaTraffic:trafficLoad];
        return self;
}
```

Figure 6.9: An object sending a 'traffic load' message

Communications between the human planners and internal *objects* take place in the same manner. Input from the human planner such as 'preferred technology' (chosen technology) which is imposed in the middle of the process is sent to the 'technology plan' *object* as shown in Figure 6.10. Human planners typically feed the input through a user interface facility so that he or she has to press a button to interact with internal *objects*.

```
@implementation MainController:Object
// A method implemented within 'main controller' object
- imposeSelectedTechnology:sender
{
        const char      *input;
        input = [[sender selectedCell] title];

        // Send preferred technology message to the 'technology plan' object
        [chosenTechnologyField setStringValue:input];
        [techPlanObj setChosenTechnology:input];
        return self;
}
```

Figure 6.10: Sending a message of the 'preferred technology'

6.1.5 Planning knowledge within object methods

The *objects* and relationships between *objects* represent the structural *knowledge* for problem solving. *Constraint knowledge* is also incorporated within the *object methods*. Therefore, an *object* may contain *methods* which derive attribute values from other attribute values or assign new parameter values of planning entities based on other parameters. The results of this calculation are also considered constraints in the planning system.

Figure 6.11 shows a commentary part of 'calcBandwith' *method*. It contains, for example, the complex mathematical formulae for calculating the equivalent bandwidth of the given traffics [53].

```
- calcBandwith:sender
/* Bandwith equivalent is calculated for all services provided at the
   particular planned area.
   It assumes a Constant for PACKET LOSS RATIO and BUFFER SIZE
   The input for the process are : #Customers, Provided Services with the
   phases of service maturity
   The RESULTS are The Bandwidth Equivalent and The Bandwidth
   Requirement
*/
```

Figure 6.11: A method containing a complex mathematical formulae

The planning rules are specified by indicating the preconditions and their triggered functions. The triggered function of a rule has the same form as the other functions within an *object class*. The execution of a function may trigger some rules, and an activation of a rule may also trigger some functions. Figure 6.12 shows a rule of thumb which is implemented within an *object* method.

```
- thisIsTransEquipType:(const char *)transEquipTypeReceived
// This message is received from the 'planned area' object
{
        if ((strcmp(transEquipTypeReceived,"Low Speed")) == 0)
                [self setMyType:"Copper"];
        else    [self setMyType:"Fibre"];

        return self;
}
```

Figure 6.12: A method containing a rule of thumb

A 'transmission system' *object* sets the type of itself according to the requirements from the services to be provided in the planned area. This function contains a simple rule which is triggered upon receiving a specific *message* from another *object*. The activation of the rule therefore triggers a function which assigns a suitable type for the transmission system according to the plan specifications.

6.2 Process control and data flow

This section describes control and function aspects of the planning system [47]. The model shows how a planning task is carried out by using all components of the system. Figure 6.13 shows a dynamic model depicting a control flow of broadband network planning by using an *event-state* diagram.

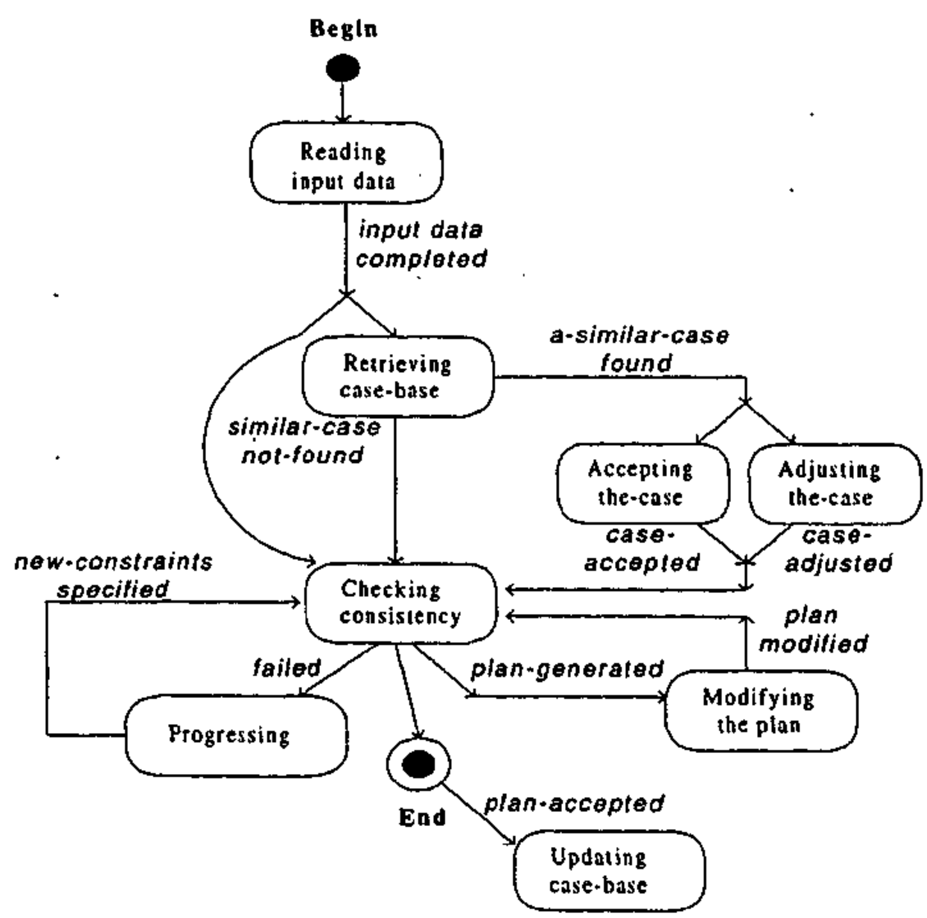


Figure 6.13: Control flow of the planning process

Network planning process begins with initial data input formulating the problem. An initial planning solution should be created as a default solution. The input data are treated as *constraints* and passed to a consistency checking process for generating a solution. At the same time, the planning software finds a solution which is consistent with the available *constraint knowledge*.

Once a plan is successfully generated, it can be either modified and go through *generate-and-modify* iterations or be stored into *case-base* as a satisfactory plan. If there is no solution found, meaning that there is no plan that does not violate one or more given *constraints*, new specifications can be defined and *attempt-and-fail* cycle continues until there is a solution found. The process described above virtually involves both the *deductive model* and *abductive knowledge*.

The use of the case-base module within the system operates as follows. A similar case (indexed in a suitable manner) from the past is retrieved in order to find a solution for the current problem case. This previous case is adjusted or accepted as a potential solution for the new situation. It should be noted that all possible solutions should go tnrough consistency checking. The option of retrieving a case-base is particularly beneficial because the past solutions might have gone through several iterations or *knowledge* intensive processes to become a satisfactory solution.

The functional aspect of network planning is modelled to show the data and computations involved in the system. A standard data flow diagram (DFD) is used for this model [47]. Figure 6.14 exemplifies a higher level network planning data flow. This figure shows three major computations namely: 'decide network technology', 'calculate traffic load' and 'dimension network'. It also describes functional relationships among input data (e.g. aggregated demanded services), output value (e.g. chosen technology) and internal data (e.g. service characteristics, technology options and area information).

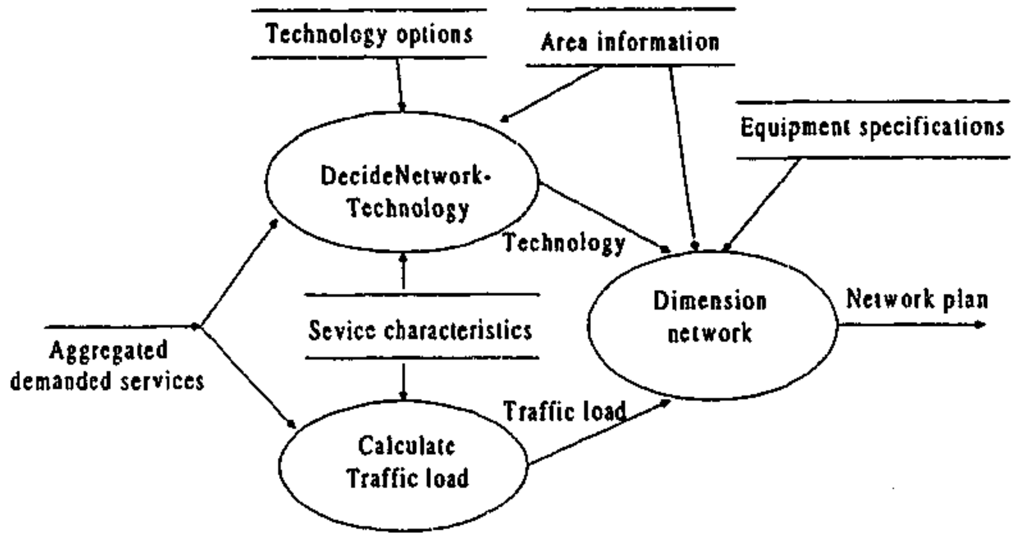


Figure 6.14: Data flow of the planning process

6.3 Input data for planning process

Figure 6.15 shows the input data panel of the planning system. The input data include general information such as total population, existing network infrastructure and the services to be provided in the area being planned.

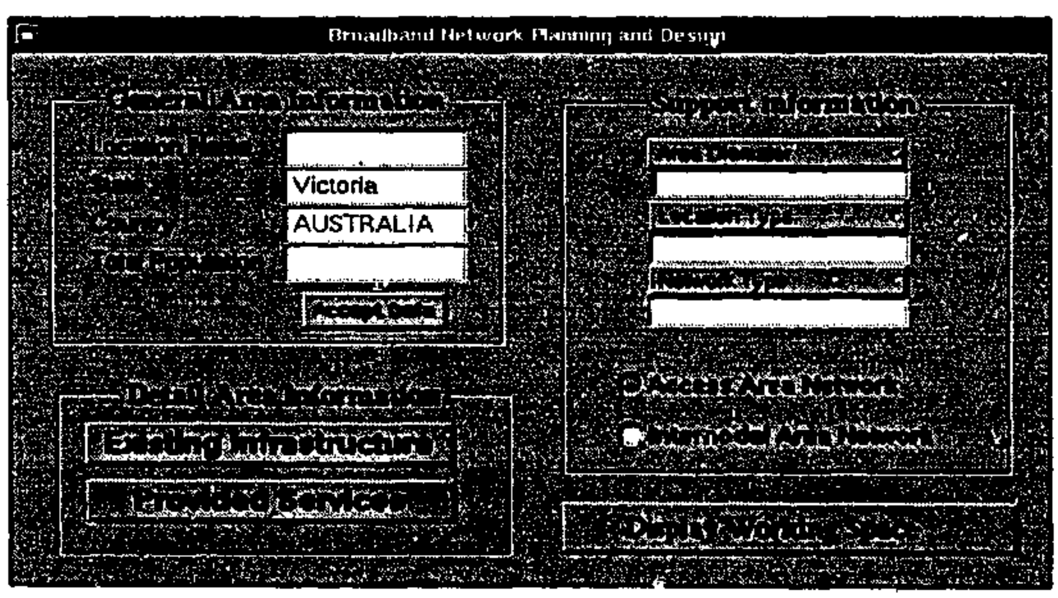


Figure 6.15: Broadband network planning input data

The system requires geographic and demographic area information such as area diameter and location type of the area. The system also needs information about the type of network being planned. It can be a local area, metropolitan area, wide area or an intercampus network.

The system solves an access area network or an internodal area network separately. However, it uses planning results of the related access area network when it deals with an internodal area network planning problem. The results of the internodal area network also become the main considerations for planning the relevant access area network.

The calculation of potential traffic offered to the network is a routine task for which network planners should not spend too much time on. The planners should concentrate more on results and spend more time in evaluating or analysing different planning solutions. The planning system incorporates methods to calculate the total traffic offered to the network. Figure 6.16 shows an example of the all services provided in a certain area being planned. It should be noted that a provision condition of each service might be different. Some services might be still in the phase of introduction and some might be still in the phase of evolution or already mature. This information is also essential for a more realistic broadband network planning.

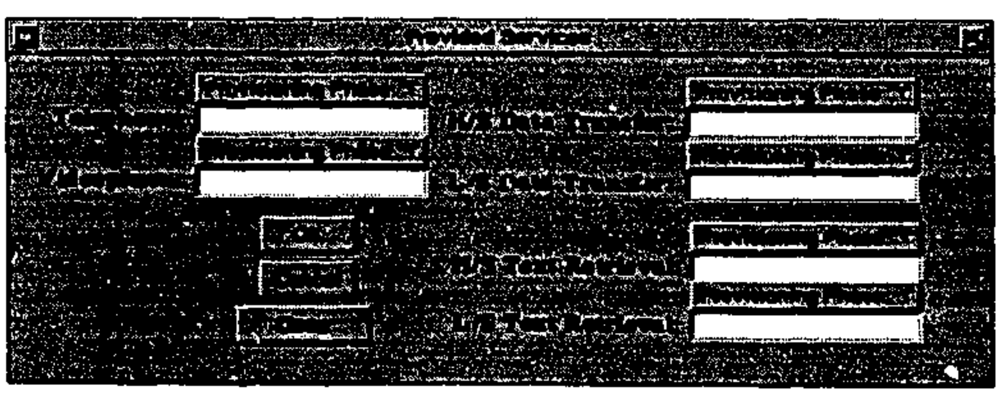


Figure 6.16: Provided services in a certain area

Figure 6.17 shows an example of the broadband network services demand. Service penetration represents a percentage of the total population that subscribes to a particular service. The busy hour call attempts (BHCA) represents the average number of attempts that a certain group of customers make use to the services at the peak hour.

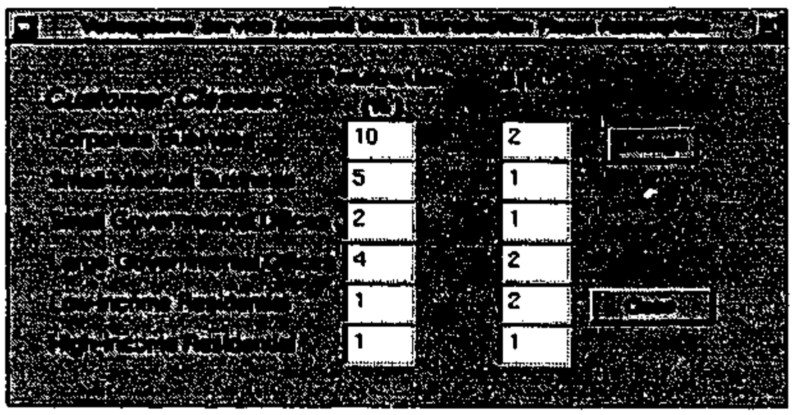


Figure 6.17: Broadband network service demands

In order to dimension the network, the system uses the traffic load estimation. The system incorporates planning *knowledge* for allocating the required bandwidth capacity based on the information about the provided services. The system converts all the traffic into bits per second by using data about service characteristics. This data is maintained and updated separately as part of the system's database. Figure 6.18 shows the service characteristics data. The system also uses all the probability factors associated with them in order to produce a network planning solution which sustains over several years to come.

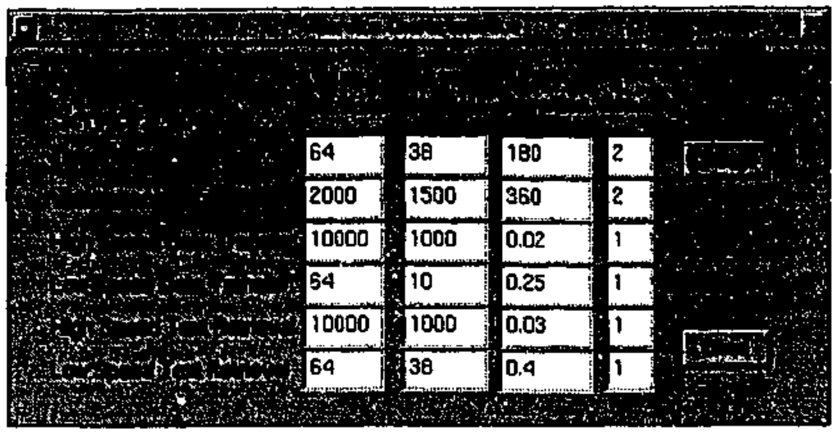


Figure 6.18: Service characteristics

Table 6.1 shows an example of traffic data for a MAN planning process. This is the result of the calculation which is performed based on the raw data fed through the system's user interface facility. Table 6.2 shows an example of traffic data for an Intercampus network planning.

Traffic types = voice and data
Traffic load = 40 Mbps

Table 6.1: MAN planning input data

Traffic types = voice, data, and video.
Traffic load = 15 Mbps (doubling every 12 months)

Table 6.2: Intercampus network planning input data

6.4 Propagating planning constraints through objects

Broadband network planning problems are *intractable* in nature because they involve many considerations and interacting factors. The best solution is selected from a large number of alternatives that are impractical to list. The complexity of the problem is the real challenge. There is no fixed algorithm that can be used for solving the problem efficiently because of the combinatorial explosion of alternatives.

This thesis implements an inference mechanism within a *constraint network* by deducing plans through the relationships between *objects*. The relationships between *objects* model the links between structural and behavioural properties of the planning entities.

Constraints include all *knowledge* which is expressed in the form of mathematical relations, rules of thumb, specific optimisation methods or procedural algorithms. A *constraint* is based on some planning rules, specific optimisation methods or some results of a database retrieval. A *constraint* can be a simple equation between two *object* attributes or between attribute and constant. All *constraints* are based on structural and domain specific *knowledge*.

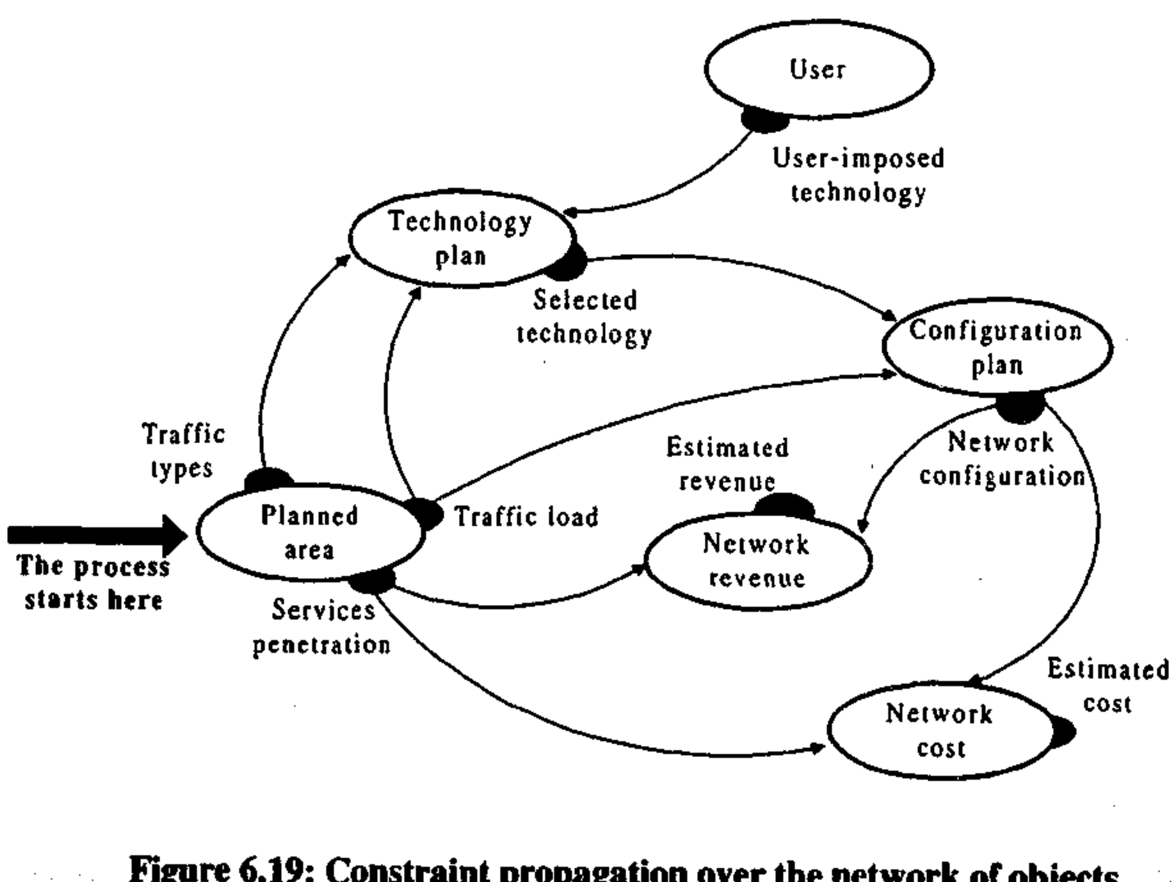


Figure 6.19: Constraint propagation over the network of objects

Figure 4.3 is reproduced here as Figure 6.19 for convenience to illustrate the operation of a *constraint* propagation mechanism in the *network of objects* for broadband network planning.

The planning system takes all information provided as well as traffic types and traffic load calculation results from the area being planned. The system firstly selects the most suitable technology according to all the constraints and takes into consideration the technologies available in the market. Figure 6.20 shows the 'technology plan' *object* from the *network of objects*.

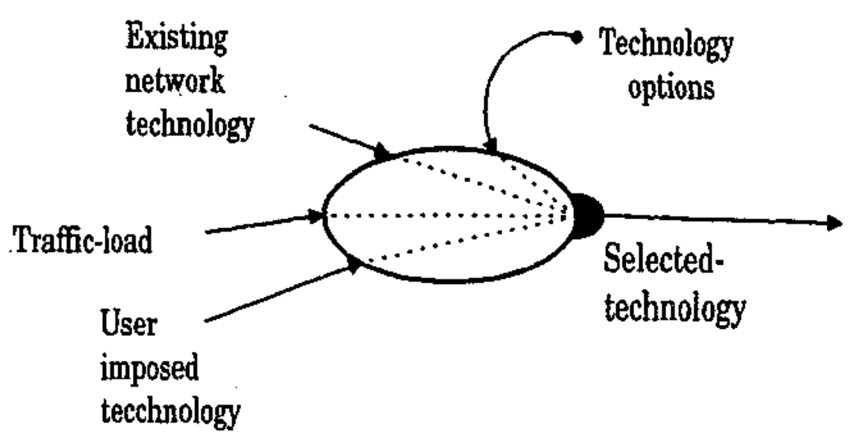


Figure 6.20: Technology plan object

It shows a planning entity with some variables relevant to the object including: existing network technology, traffic load, selected technology, and network technology options. Possible values of the network technology are: POTS, ISDN, Frame Relay, DQDB-MAN, and ATM [15].

The network technology options is part of the data incorporated in the system's database. This system maintains tables of technology options for different types of broadband networks. The data are kept separately so that change, addition or subtraction of the list contents can be done whenever required. Figure 6.21 shows a list of the technology options kept in the database for a MAN. The list of technology options for an Intercampus network is shown in Table 6.3.

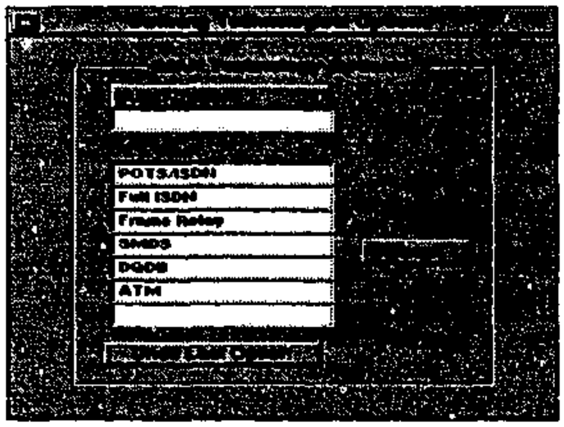


Figure 6.21: MAN technology options

1.Public carrier's ISDN
2.Public carrier's Frame Relay
3.Public carrier's ISDN/ Frame Relay
4.Public carrier's ATM
5.Microwave Radio TDM: 34 Mbps
6.Microwave Radio ATM: 34 Mbps
7.Leased line TDM
8.Leased line ATM

Table 6.3: Intercampus network technology options

The 'technology plan' *object* has an attribute called 'selected technology' whose value is constrained by the a number of variables from different *objects*. 'Existing technology', 'traffic load' and 'services penetration' data of the planned area are *constraints* for the 'selected technology' value. The 'selected technology' is also constrained by a preference of the user (planner) who might attempt to impose a particular technology in the planning process. The *code* in Figure 6.22 shows the propagation of the 'traffic load' *constraint* from the 'planned area' to the 'technology plan' *object*.

```
- setTrafficLoad:(int)theTrafficLoad
{
        if (theTrafficLoad) {
                trafficLoad = theTrafficLoad;
        }

        // Send 'traffic load' to the 'technology plan' object
        [tPlanObj areaTraffic:trafficLoad];

        return self;
}
```

Figure 6.22: Propagating a traffic load constraint

6.4.1 Deductive process

The planning system deduces facts from other facts by propagating properties in terms of *object* attribute values through the relationships between *objects*. For example, a 'traffic load' *constraint* propagates to the 'technology plan' *object* and then this fact will be used to deduce new facts. A *constraint* propagates to all relevant *objects* for the planning. It propagates also to the lower level *objects* within any hierarchical structure of the *objects*. This *deductive process* uses the *network of objects* as the means to reach a conclusion. It searches over the *constraint-based model* as follows:

(1) Select a possible solution;

(2) Propagate results to the relevant *objects*;

(3) Accept the solution if the results are 'consistent' and continue from (2);

(4) If inconsistency occurs, repeat the same process from (1).

The next possible solution is checked when the current value is not consistent. The selected technology is chosen from all possible values. Figure 6.23

illustrates the generation of the possible solutions. A value which is consistent with all the constraints at this stage is then propagated to the relevant *objects*. The *constraint propagation* process continues until it reached a final planning solution which is consistent with all the *constraints*.

POTS ← First possible solution
ISDN ← Second possible solution
Frame Relay ← Third possible solution
DQDB-MAN ← Fourth possible solution
ATM ← Fifth possible solution

Figure 6.23: Generating possible solutions

6.4.2 Abductive process

The planning system gains the power of inference mechanism by propagating *constraints* within the *network of objects*. New attribute values or facts are derived through the application of numerical computations upon base attributes. Deriving new values is also performed by exploiting the power of logical inference which is implicitly available in the *object-oriented* paradigm. The execution of *object* functions cause changes in attribute values and the changes in attribute values cause execution of the functions. *Constraints* operate both within an *object* and span across multiple *objects*. Internal *constraints* can be seen as the property of an *object* like attributes and functions. They will be triggered when certain parameter values are updated or functions are executed.

Objects provide a much more powerful inferencing system than the *rule-based* system because of the richness and robustness of the *object-oriented* approach. The features of *objects* such as abstraction, encapsulation and, inheritance have the ability to build a rich taxonomical hierarchy of real-world entities and model the relationships between them very clearly. This is something that most logic programming environments do not support [51].

This planning system enforces consistency in order to reduce the solution space by using multiple-sources of *knowledge*. This thesis calls these different types of *knowledge* as *constraint knowledge*. Each *constraint knowledge* is very specific and each of them is based different types of knowledge which is only suitable to a specific task.

The system uses the *constraint knowledge* in the first place to reduce the number of feasible solutions and then exercises the *deductive process* for generating each feasible solution. Therefore before a *deductive process* takes place, the following procedure applies for each *object*:

(1) Apply *constraints* to reduce the number of feasible solutions.

The set of *constraints* include:

- Selected technology = function of (Traffic load, Existing network technology)
- Selected technology = function of (User-imposed technology)
- Selected technology = function of (Corporate policy for network technology)
- Selected technology = function of (Cost of the resulting network configuration)
- Selected technology = function of (Business profit target)

The planning system uses all the *constraints* applied to the technology decision so that the feasible solutions are reduced to only 'DQDB-MAN' and ATM. Figure 6.24 shows two feasible values after applying the *abductive knowledge* implemented within the system. The process then continues as follows:

(2) Select a possible solution;

(3) Propagate results to the relevant *objects*;

(4) Accept the solution if the results are 'consistent' and continue from (3);

(5) If inconsistency occurs, repeat the same process from (2).

- DQDB-MAN ⟵ First possible solution
- ATM ⟵ Second possible solution

Figure 6.24: Reduced possible solutions for the network technology

The complete process described above should clarify distinction between planning with a pure *deductive* system and the use of *abductive knowledge* of this planning system. In the pure *deductive* system, a solution generation takes place for the first stage and it is followed by a consistency check. This is good for the situation where the domain *knowledge* is very limited. On the other hand, our system applies *constraint knowledge* at the beginning of the planning process and solution generation follows after the number of feasible solutions has been significantly reduced. By applying *constraints* from the *abductive knowledge*, we avoid exhaustive search and combinatorial explosion.

This thesis uses a hierarchy of *constraints* in order to avoid conflicting rules, which frequently occurs in many *knowledge-based systems*. Therefore our system does not need any conflict-resolution strategy where multiple, potentially conflicting rules, exist simultaneously. The hierarchy is reflected in the internal structure of the *constraint knowledge* as well as process control flow. The mechanism to ensure a *constraint knowledge* ordering in this planning system is as follows:

Constraint knowledge ordering:

- The system uses the most important *constraint knowledge* first when reducing a number of possible solutions. The order of all the *constraint knowledge* is maintained.
- The system uses a flag system for the *constraint* variable to indicate a *hard constraint.*

Variable ordering:

- The most critical *constraint variable* is checked first.
- *Constraint* propagates from the most critical variable to the less critical one.
- Backtracking, if necessary, is performed in the order of the less critical variable to the most critical one.

In this context, the system takes human planner preferences as *hard constraints* which should be met whenever possible in this system. Otherwise, the system reports that there is no solution meet the provided constraints.

Possible values for the selected technology are reduced based on the *constraint knowledge*. The previous section has discussed how some *constraint knowledge* used for reducing the possible number of network technology options for the planned area. From the reduced alternatives, one value is generated at one time and the current value of the technology plan *object* becomes a *constraint* to the relevant *objects* e.g. 'configuration plan' *object*.

The planning system also maintains a typical specification of network elements required for each technology option. Figure 6.25 shows a specification of the network requirement for DQDB-MAN configuration. Figure 6.26 shows a detailed specification, which includes capacity and price of the equipment, of a piece the network equipment. These kind of data are made available within the system for planning broadband networks.

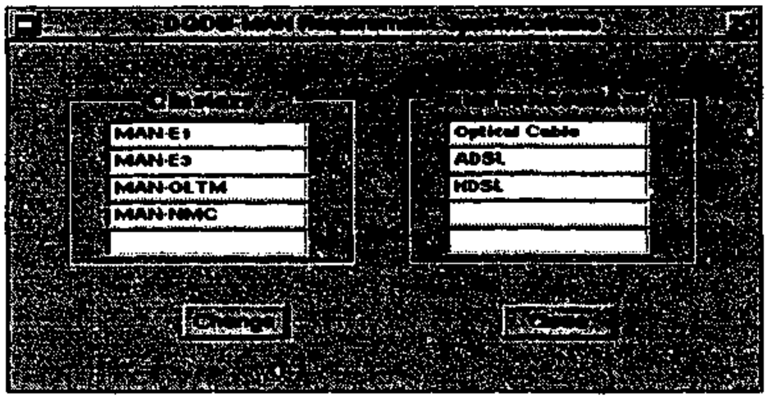


Figure 6.25: DQDB-MAN requirement

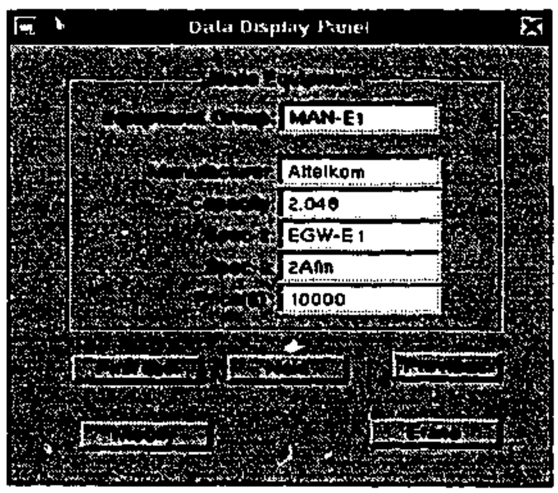


Figure 6.26: Equipment data example

The 'configuration plan' *object* uses a technology selection result of the 'technology plan' *object* as well as a traffic load value of the 'planned area' *object* as *constraints* to determine a network configuration. The system determines the network dimension such as type, quantity and capacity of the all network elements needed. The result of network configuration is then also propagated to other relevant *objects* and so on.

Set of *constraints* for network dimensioning:

- Network configuration = function of (Selected technology, Traffic load)
- Network configuration = function of (Existing infrastructure)
- Network configuration = function of (Investment plan)
- Network configuration = function of (Rolling-out plan)

The planning system calculates a network profit by calculating optimal cash flows which involves calculation of net cash possibly generated by the network. This method examines the cash flows occurring at *different points in time*. Cash flow calculation includes all cost components and predicted revenue as well as other factors based on financial factors. All cost components such as equipment, installation, upgrade and management cost are to be taken into account in order to have more precise cash flow results. Network cost and its predicted revenue are the *constraints* used to decide the optimal plan in the last stage.

Set of economic *constraints*:

- Optimal network plan = function of (Network cost, Network predicted revenue)

The revenue from all services of each scenario is calculated. Service penetration scenario is predicted customer penetration for a set of services over a certain period of time. There are some service penetration scenarios available from several different market surveys. Each scenario has a probability factor which indicates the probability of each scenario to occur. The scenario contains a set of services to be provided complete with their predicted penetration. Therefore, a potential revenue can be calculated based on this scenario.

However, this thesis does not use a complete financial model to determine the network *profitability*. It only uses a simple means for evaluation and comparison of profitability as an indicative measure of different solutions.

6.4.3 Evaluation of possible solutions

The propagation of the *constraints* over the *network of objects* throughout the planning process is a two-way *constraint propagation*. The application of abductive knowledge practically ensures the correct decision is taken along the step by step process of the planning in one direction (forward direction). Each value assigned and then being propagated from one *object* to the others is guaranteed to be satisfactory. Therefore the number of *backtracks* which take place within the solution finding process is reduced to a minimum. However, the propagation of results in terms of assigned values, after an *object* receiving any *constraint* from other *objects*, is sometimes necessary. Figure 4.11 is reproduced here as Figure 6.27 to illustrate this mechanism.

The values of estimated revenue and cost are backwardly propagated by 'network revenue' and 'network cost' *objects* respectively to the 'configuration plan' *object* after the first two receive a *message* containing a network configuration *set of value* from the latter. A *value* of estimated profit is also propagated by the configuration plan *object* to the technology plan *object* after receiving a selected technology *message* from the latter and so on.

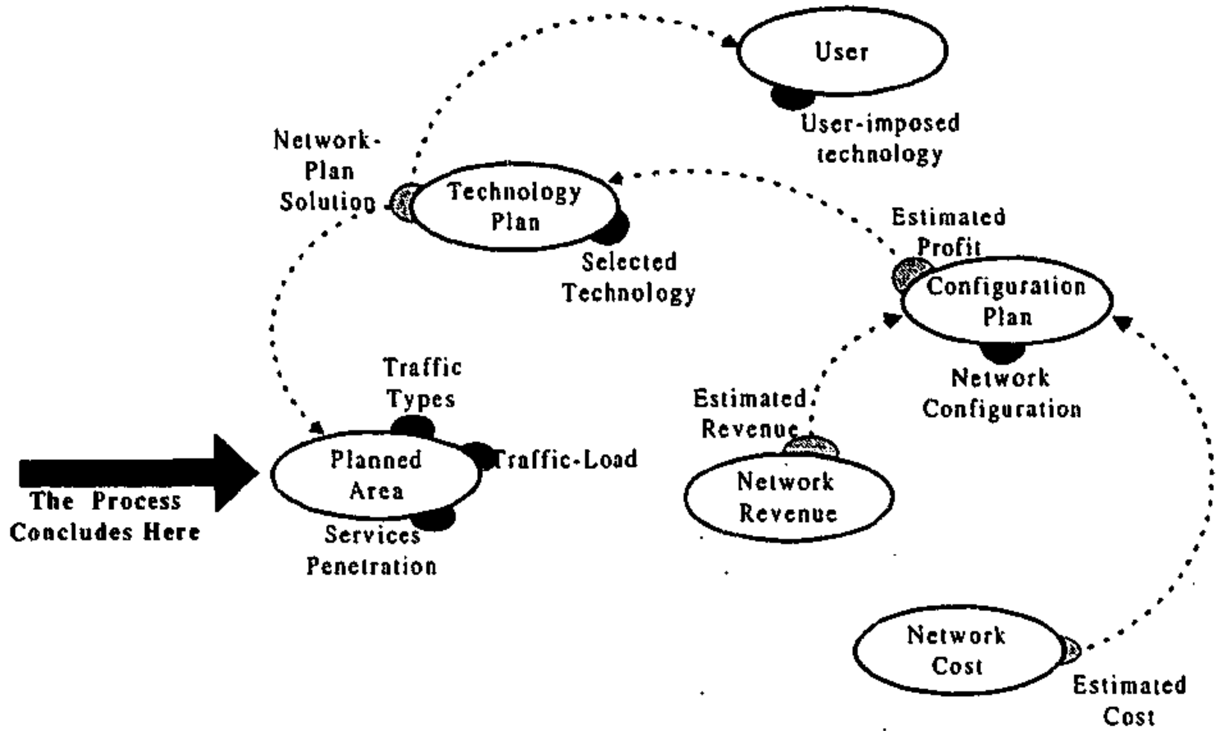


Figure 6.27: Backward propagation of partial results

This planning system uses the backward propagation process to find a local optimum solution within the perspective of each particular *object*. Backward propagation of the partial results enables the system to compare among the reduced possible values after the application of *abductive knowledge*. This system uses backward propagation of partial results to find an optimal solution between the two alternatives. The solution-finding that consists of both forward and backward propagation will take much longer time if it is applied to a larger number of possible solutions. This is why the *abductive knowledge* is essential to this planning system.

The planning system produces a network planning solution in terms of the chosen technology and all distinct elements required to satisfy the planning requirements. Table 6.4 shows one example of the planning solution for a MAN. It includes facilities required at both the internodal and access levels of the network.

Table 6.5 shows a solution for the Intercampus network.

Selected technology = SMDS on DQDB-MAN		
Network configuration:		
	Internodal level =	4 x E3 DCS
		4 x 2 x OLTM
		35 km x Optical Cable
		1 x Network Management Centre (NMC)
	Access level =	14 x EGW-E1
		70 km x HDSL Link

Table 6.4: MAN planning solution

Selected technology = Microwave ATM 34 Mbps	
User application networks:	- shared/dedicated 10 Mbps Ethernet
	- 25 Mbps ATM
	- 155 Mbps ATM
Core network: ATM cell switching	
Network Elements:	7 x 2 ATM switches
	ATM workgroup switches
	7000 Routers
	AGS+ Routers

Table 6.5: Intercampus network planning solution

6.5 Conclusion

This chapter has discussed *objects* and shown how the broadband network planning problem structure is modelled in terms of the *network of objects*. It has discussed the structure of *objects* and the implementation of the inter-objects relationships. The chapter has also shown how the *constraint knowledge* is incorporated into *objects* and how to propagate such *constraints* over the *network of objects*.

This chapter has further elaborated how the combination of the *deductive process* and *abductive knowledge* works together. It has also provided a control and data flow of the software system in terms of event-state diagram. Finally, the chapter has discussed a step-by-step planning process through examples.

Chapter 7

Evaluation and Conclusion

This thesis has presented a *constraint-based* system for planning broadband networks. It has developed a software which integrates a *constraint-based model* with different types of *knowledge*. The system applies a *deductive process* within the *constraint-based model,* and uses different types of knowledge *abductively* to find the planning solutions. This chapter examines important aspects of the design and implementation of the planning system. This chapter also provides conclusions of the thesis and suggests topics for further research.

7.1 Software design analysis

7.1.1 Modeling the planning entities

Modeling the planning entities in terms of objects and the relationships between them is the central idea of this broadband network planning system. Models of the planning objects and their functions give a great advantage in maintaining the system. This approach gives flexibility and ease for addition, subtraction, modification and replacement of the models.

The object model of each broadband network planning entity is very simple compared to the actual entity. The models do not necessarily capture every detail of an actual entity. The abstraction of the entities only represents the attributes and functions which are worth representing. This abstraction adequately represents the actual entities to carry out the planning task.

Modeling of the planning objects should start from the top level objects. In this thesis, 'planned area', 'technology options', 'network services', 'network plan', and 'network equipment' are examples of them. The relationship models between objects is firstly defined in a very broad way. Modeling objects and relationships between objects are continuously refined until the smallest objects are reached according to the granularity required for specific problems.

In particular, the same approach applies for modeling the telecommunications network object in the area being planned. Modeling a telecommunications network and relationships between the telecommunications networks is depicted as a network hierarchical structure shown in Figure 7.1. A national network consists of several metropolitan area networks, and a metropolitan area network also embodies several local area networks. Traffic between two networks is the main link to represent the relationship between two networks. Each network is modelled as an object which may consist of several smaller objects of the lower level networks. Planning each network involves models of the 'planned area', 'technology options', 'network services', 'network plan', and 'network equipment' as described in a previous chapter.

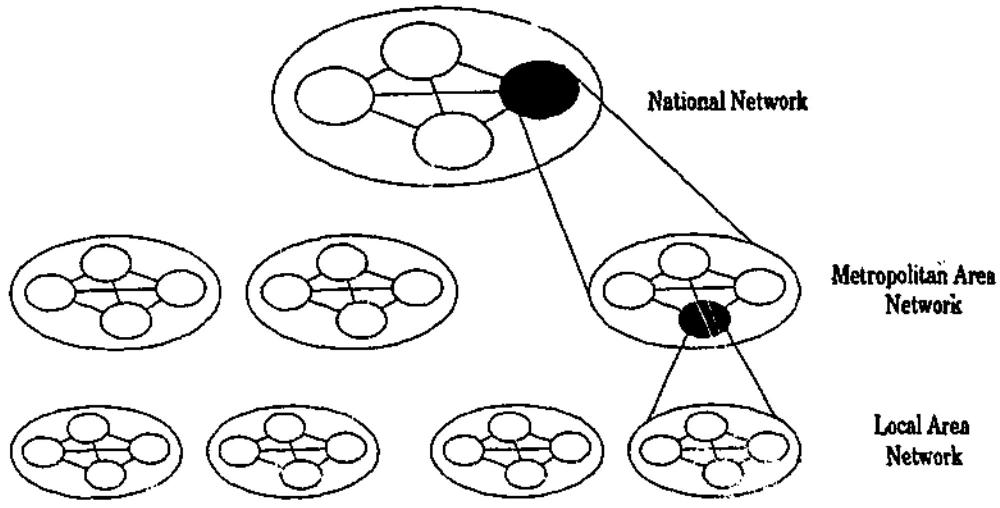


Figure 7.1: Hierarchical telecommunications networks as an object

7.1.2 Constraint-based model

This thesis models all possible choices in planning. Therefore, the model deals only with discrete values of the variables. The system views the numerical optimisation result as an output from the *abductive part* of the system. This method supplies its results to the *constraint-based model* for further processing. This seems to carry some limitations. But this simplification is found reasonable to provide satisfactory planning solutions. It also prevents the *deductive process* from being computationally too expensive.

The model ensures the system finds a planning solution for any planning specification or input data provided. The system accepts a change of the planning specification if the current one does not lead to any satisfactory solution. The idea is to realise a spread-sheet like planning system which can accept any change of any variable and show the consequences of each change.

Building a *constraint-based model* is a continuous process. Therefore it is perfectly suited to the dynamic nature of the broadband network planning problems. Model refinement can be done throughout the life of the system.

7.1.3 Constraint knowledge

The different types of *constraint knowledge* used in the system improve the computation speed. It also gives a procedural meaning to the planning. *Constraint knowledge* seems to cover all possibilities of the planning when the size of the problem is small enough. But, for a complex system, like broadband networks, it is hardly possible to have a complete coverage of the *constraint knowledge*.

The significance and power of the *deductive model* is minimal for simple problems. In this situation, the problem is simple enough so that the *abductive knowledge* is nearly complete. The role of the *deductive model* increases with the increase of the problem complexity. However, application of *abductive knowledge* at each stage of the planning process reduces the complexity of the *deductive processing* required to solve the planning problem.

7.2 Planning solution

The output of the system is a planning solution which should be optimal according to the available domain *knowledge* incorporated into the system. However, it is difficult to evaluate the performance of the system in terms of the quality of the solution. This thesis does not make any comparison between the solution it provides with that from others. There is not enough information from other systems available to make any fair comparison with this system. This is the main reason such a comparison is not included in this thesis.

7.3 Computation time

This system aims to be an interactive planning software which should be robust enough to give at least one solution given any planning input data. Therefore, an adequate *deductive model* is required to generate all possible solutions. However, the speed of computation is a real problem. Because of the search, the system produces many alternatives and grows exponentially according to the number of planning entities. The planning system reduces the search space by using the *constraint knowledge*. This method coupled with the *constraint-based model* increases the speed significantly while retaining its robustness.

The system only produces a single solution at one time. This seems to carry some limitations. However, the system still has the capability to produce a complete and correct plan. Furthermore, this approach has been useful to overcome the combinatorial problem in the planning.

The use of *abductive knowledge* significantly increases the speed. However, although the *abductive knowledge* is expected to be always correct, it is not always successful in providing a solution. This is because the *abductive knowledge* is unlikely to be complete for tackling all possible situations. Therefore, the system will fall back to the *deductive process* when the *abductive knowledge* fails to provide any 'direction' for the search. Consequently, it can add on a computational overhead to the whole process.

Figure 7.2 shows a simulation of the time required to find a planning solution by using a pure *deductive process*, a pure *abductive process*, and a combination of both of them. The planning system is implemented in an Intel-based PC Pentium 90.

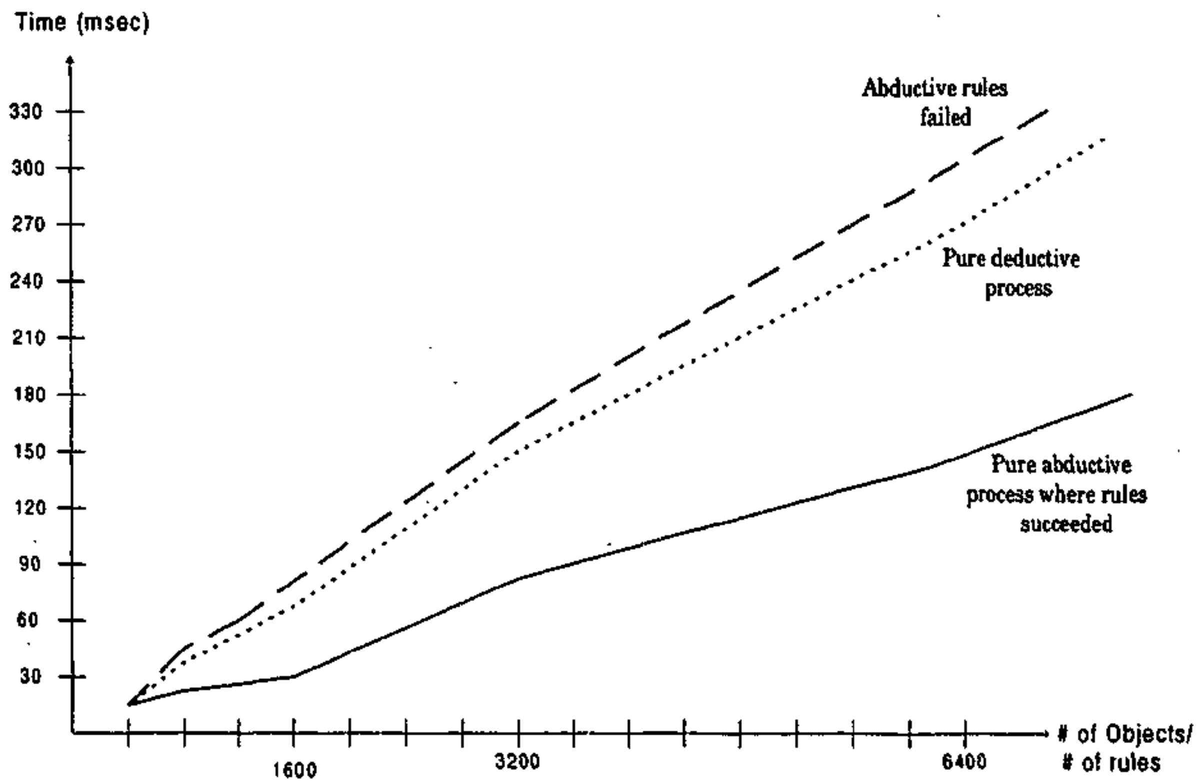


Figure 7.2: Computation times of the planning processes

The number of objects involved in the planning process is represented on the x-axis. The *deductive process* is applied to the number of objects which gradually increases. For this purpose, the number of objects involved is multiplied by making an increasing number of copies of them. The number of rules involved in the *abductive process* is also multiplied in the same way. The number of rules involved in the *abductive process* is also represented on the x-axis.

The dotted line in Figure 7.2 indicates the time required for finding a planning solution by using purely *deductive process* and the solid line indicates the time required for finding a solution by applying *abductive rules* only. The time required for the planning process that involves a small number of objects and *abductive rules* is very small. The difference between the time required for a purely *deductive* and *abductive processes* becomes more significant when each of them involves more than 40 objects and rules. Figure 7.2 shows that the reward for using *abductive rules* clearly increases as the size of the problem increases. All other types of *abductive knowledge* of the system also give the same advantage because they all contribute to the solution-finding in the same fashion as the *abductive rules* do.

The dashed line in Figure 7.2 shows the time required when *abductive rules* fail to find any solution. This situation occurs when there is no single rule of the *abductive rules* applicable for the current planning specifications. The system then uses the *deductive process* over the constraint-based model to find a solution. Additional time is required for finding a solution but this approach certainly makes the planning system more robust. Moreover, this situation is expected to be an exceptional situation or at least does not occur frequently.

Processing a *constraint network* is a highly expensive process. Processing a *constraint network* using chronological *backtracking* results in $O(k^n)$ time complexity [20]. Chronological *backtracking* is a very inefficient method. The best results of the *backtracking* method proposed is $O(e^n)$ time complexity by using the so called 'intelligent' *backtracking* method [63]. The time required to process a *constraint network* increases exponentially with the increase in the number of objects. However, Figure 7.2 shows that the time required to deduce solutions within this *constraint-based model* seems to increase linearly. This encouraging result is obtained due to embedding *constraints* within the *constraint-based model.* This result also suggests a potential capability of this system to be scaled up for dealing with a very large planning problem.

7.4 Conclusion

This thesis has described the broadband network planning tasks. It has shown the complexity of the problems and the urgency for the availability of softwares to deal with the problems. The complexity comes from the many interrelated factors involved, the rapidly changing environment and the semi-structured nature of the problems. Therefore any system which attempts to solve the problems in a fragmented manner is not adequate. The required system should be able to incorporate analytical methods as well as subjective methods of the experienced planners which reflect human expertise.

This thesis has presented a design philosophy and a software for carrying out the broadband network planning tasks. It has proposed a new software architecture that provides a framework for dealing with the highly complex problems in a more comprehensive way. The prototype of the system has been implemented to demonstrate the power of the *constraint-based* approach in the domain of broadband network planning. The thesis has successfully developed *deductive models* for parts of the Metropolitan Area Network (MAN), Intercampus network, and ISDN access network planning cases. The models are sufficient to test the system although further work is still needed to make it ready for more practical uses.

The *constraint-based model* developed in this thesis has made modeling the relationships of the planning entities easier and highly manageable. Modeling relies mainly on the important functions of each entity and its relationships with relevant entities. It avoids the necessity to enumerate exhaustively *if-then* relationships of the planning. The *deductive model* may not be very efficient but it has ensured that any resulting solution is satisfactory and does not omit any possible solution. Therefore it is robust and provides a high degree of confidence for the user.

Different types of *abductive knowledge* have been used in this planning system. *Abductive knowledge* has made the system very efficient. However, it is unlikely to achieve a complete coverage for such broadband network planning applications. However, this thesis has successfully put both the deductive models and abductive knowledge together to gain efficiency and robustness. A combination of them has proved to be effective to produce a robust planning system.

This work has successfully achieved its targeted research goals, namely to design and implement a new software architecture for a planning system; to build an innovative application of *constraint satisfaction techniques* in the domain of broadband network planning; to develop a *constraint-based model* for the broadband network planning problems; and to incorporate all planning knowledge within a *network of objects*.

7.4.1 Further work

At the end, this work has suggested further studies to improve the planning software. Continued study on modeling the planning entities is needed to realise a more viable planning system for broadband networks. Further research for building a software interface for the network planners/users to create and update the *deductive models* would be very useful.

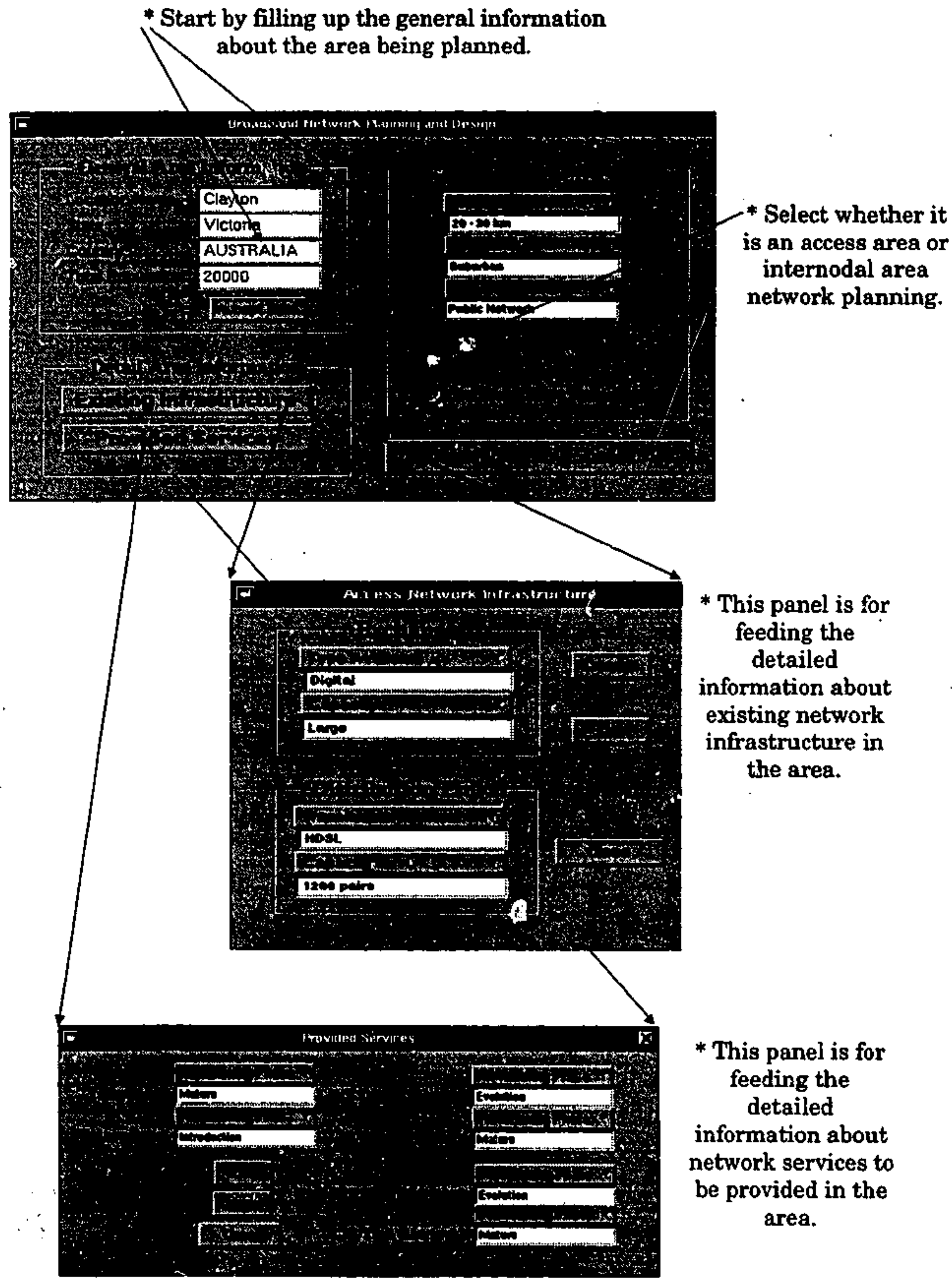
The current system has demonstrated that the *case-base* module can significantly improve the performance of the system. However, further work on developing a realistic scheme for *indexing* the broadband network planning cases and defining *similarity* among the cases is clearly required.

The performance simulation of the resulting networks produced by any planning system is highly desirable. This work has also left this area for further research. A completely new simulation module can be added to the current software or any existing network performance simulation tools can be interfaced to this system.

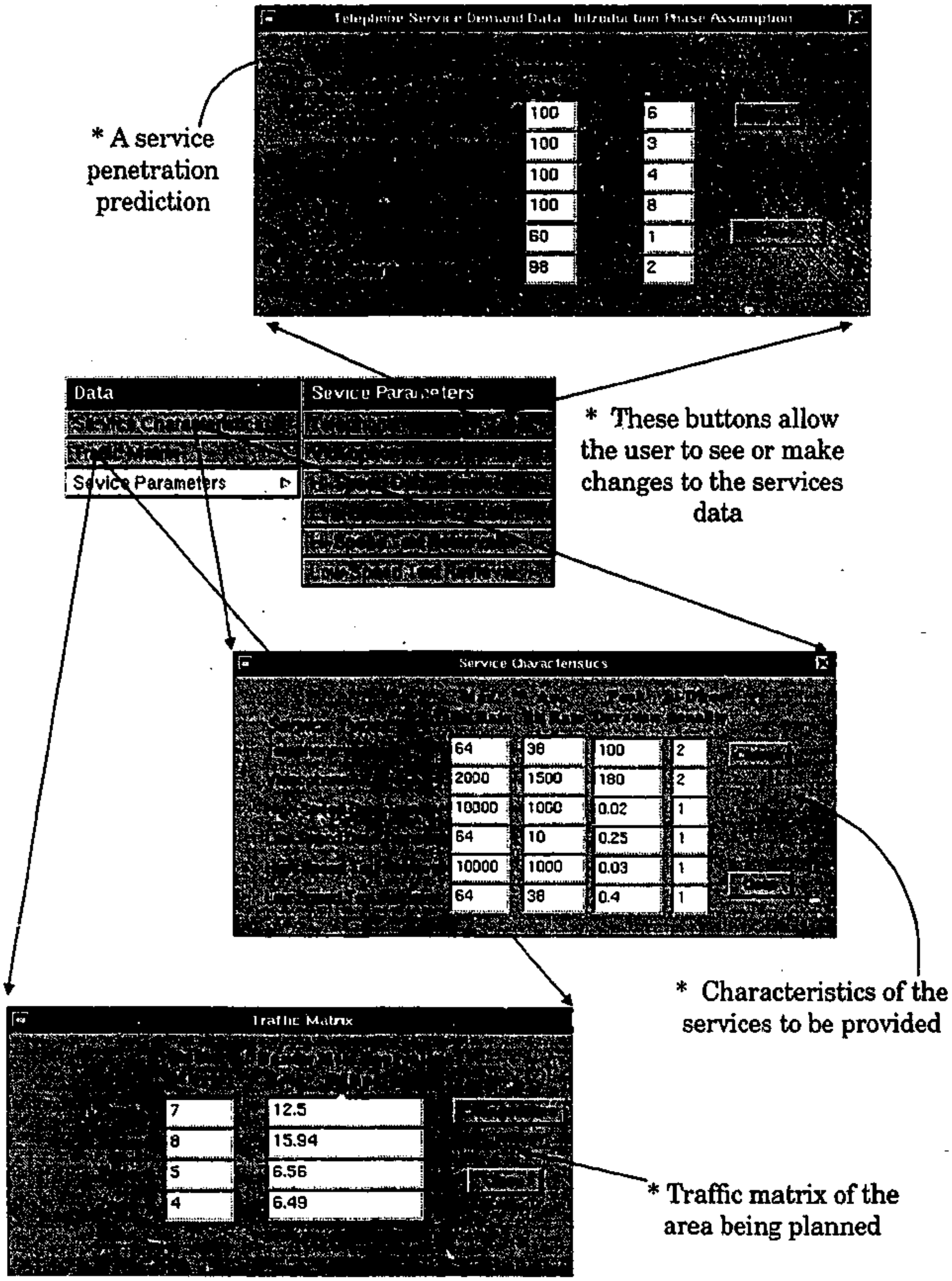
Appendix-A

Snapshots From The Planning and Design System

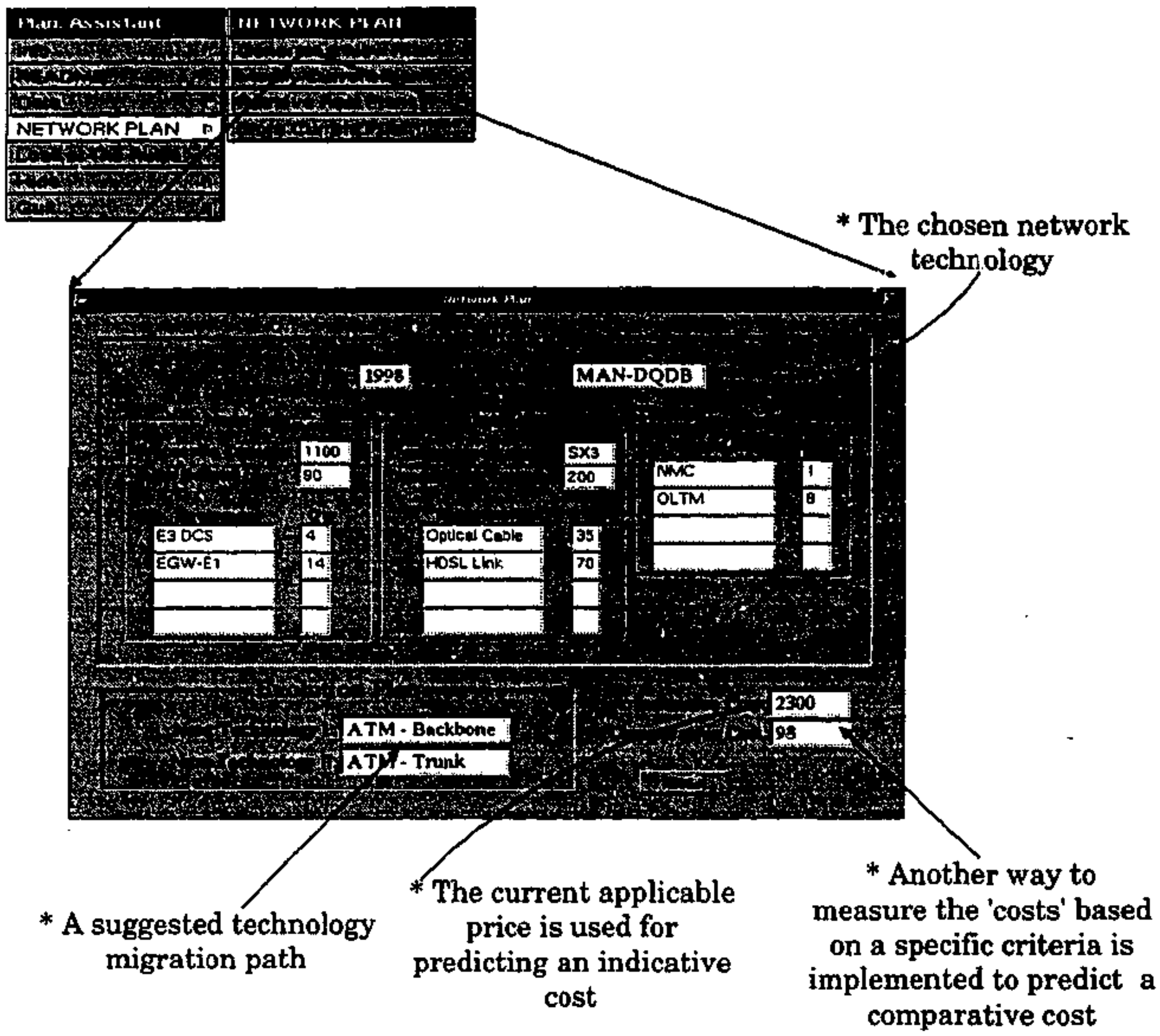
The following guide describes how to use the planning and design system. A series of snapshots of the system operation with step-by-step actions, panels and brief explanations are shown as follows.



Calculation of the total traffic load offered to the network is essential at initial stages of the process. It uses data about the prediction of the services penetration in an area being planned. Below are examples of telephone service penetration data and service characteristics. A traffic matrix result of the area is shown at the bottom of this page.

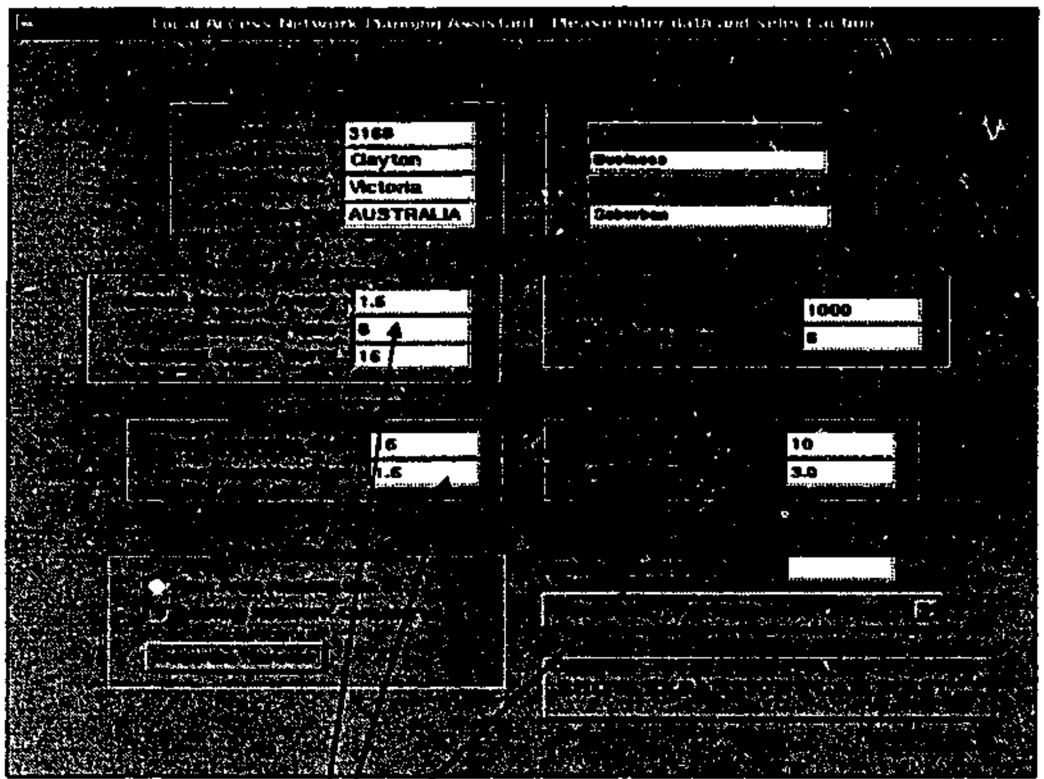


The user can generate a new plan (solution), make some changes to the current solution and see the result, or store the current solution into the case-base. The user can also see solutions from the past (if they are available).



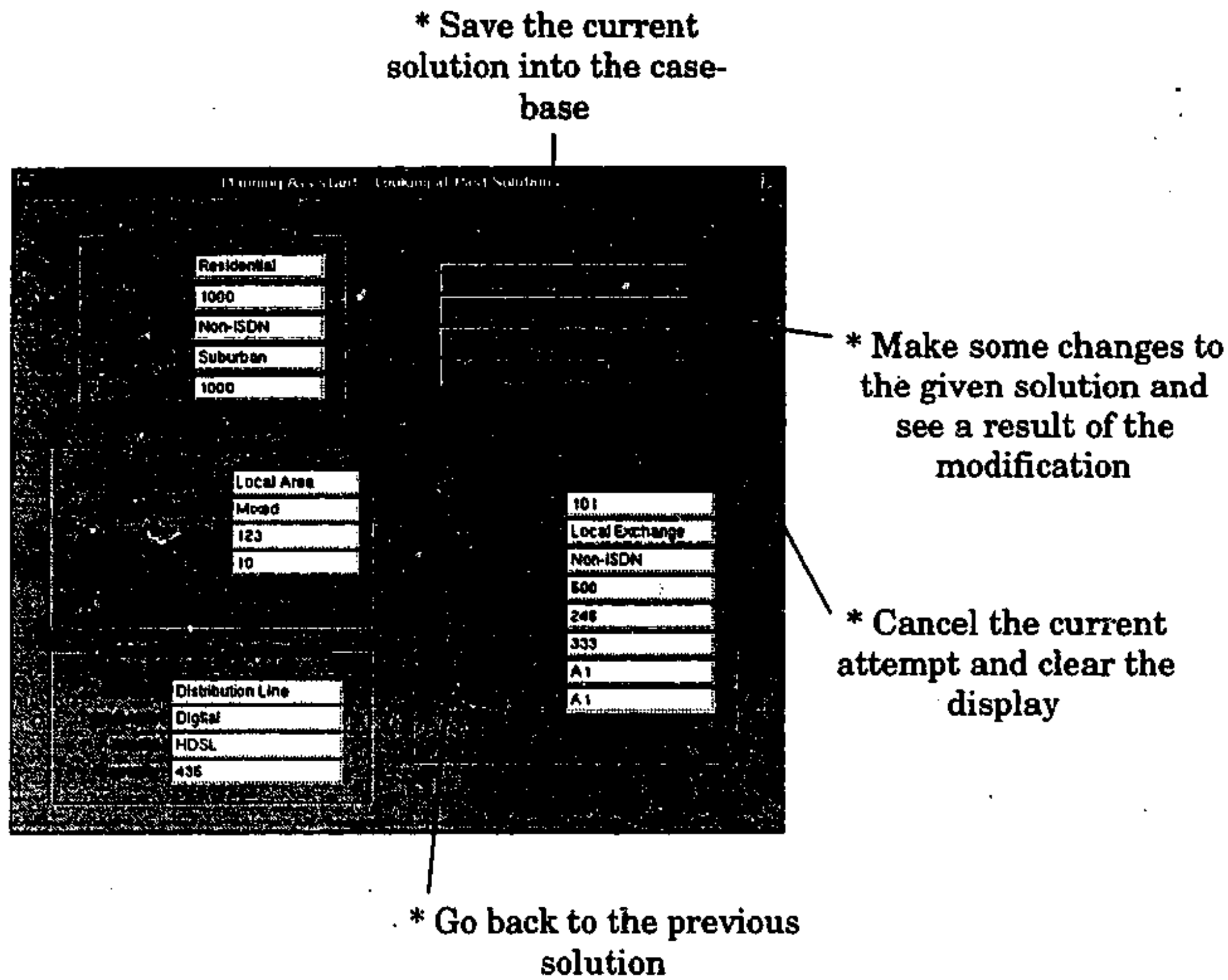
The network solution contains a selected technology with related configuration as well as a suggested evolution plan for the network technology. The system also provides total indicative costs and comparative costs for each solution.

The following figure shows a planning and design problem with initial input data and some economic factors to be considered.



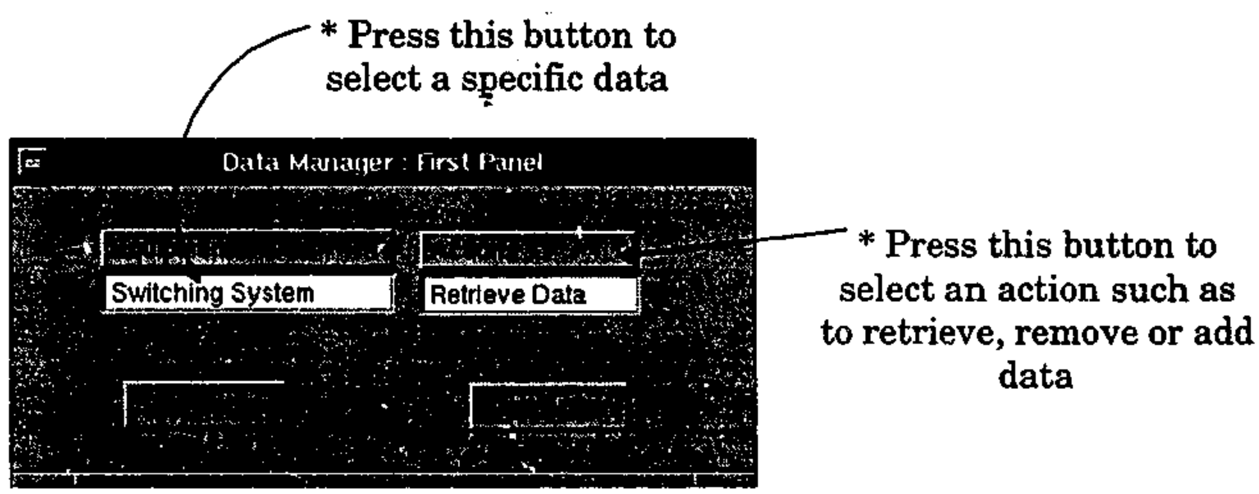
* Some economic parameters for optimizing an investment plan in the local access area network

The following page shows a snapshot of a solution provided by the system for the previous case. The solution is found by retrieving solutions from the case-base. The user can use a solution from the past as an initial step to find a more satisfactory solution.

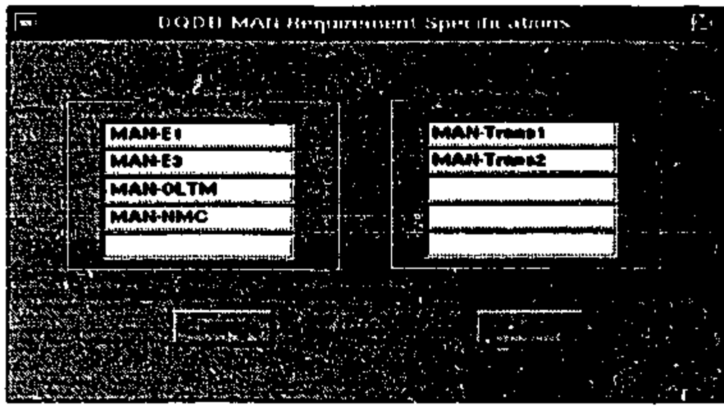


The database system is virtually integrated to the whole planning and design system.

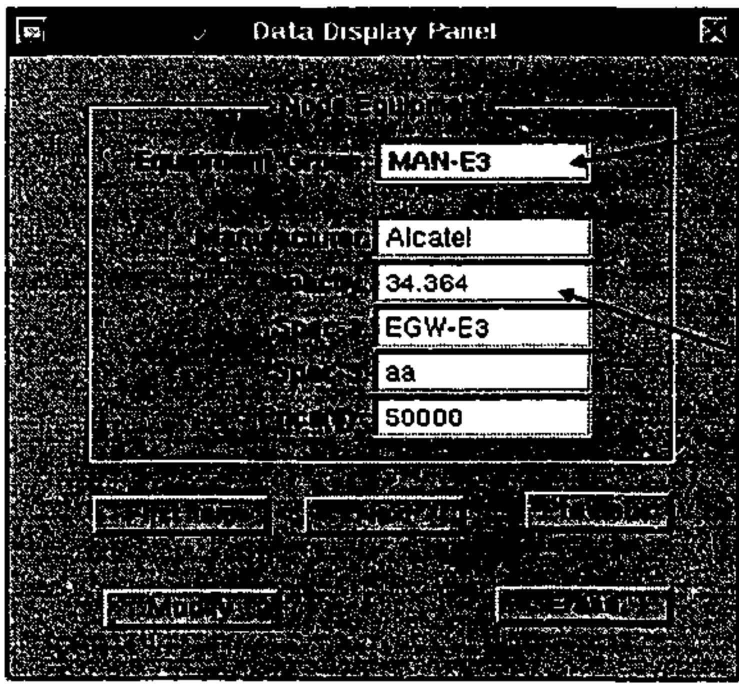
The following is the main panel of the database system.



Following is a required elements for DQDB-MAN network. The same kind of data for other network technologies are maintained in the system's database.



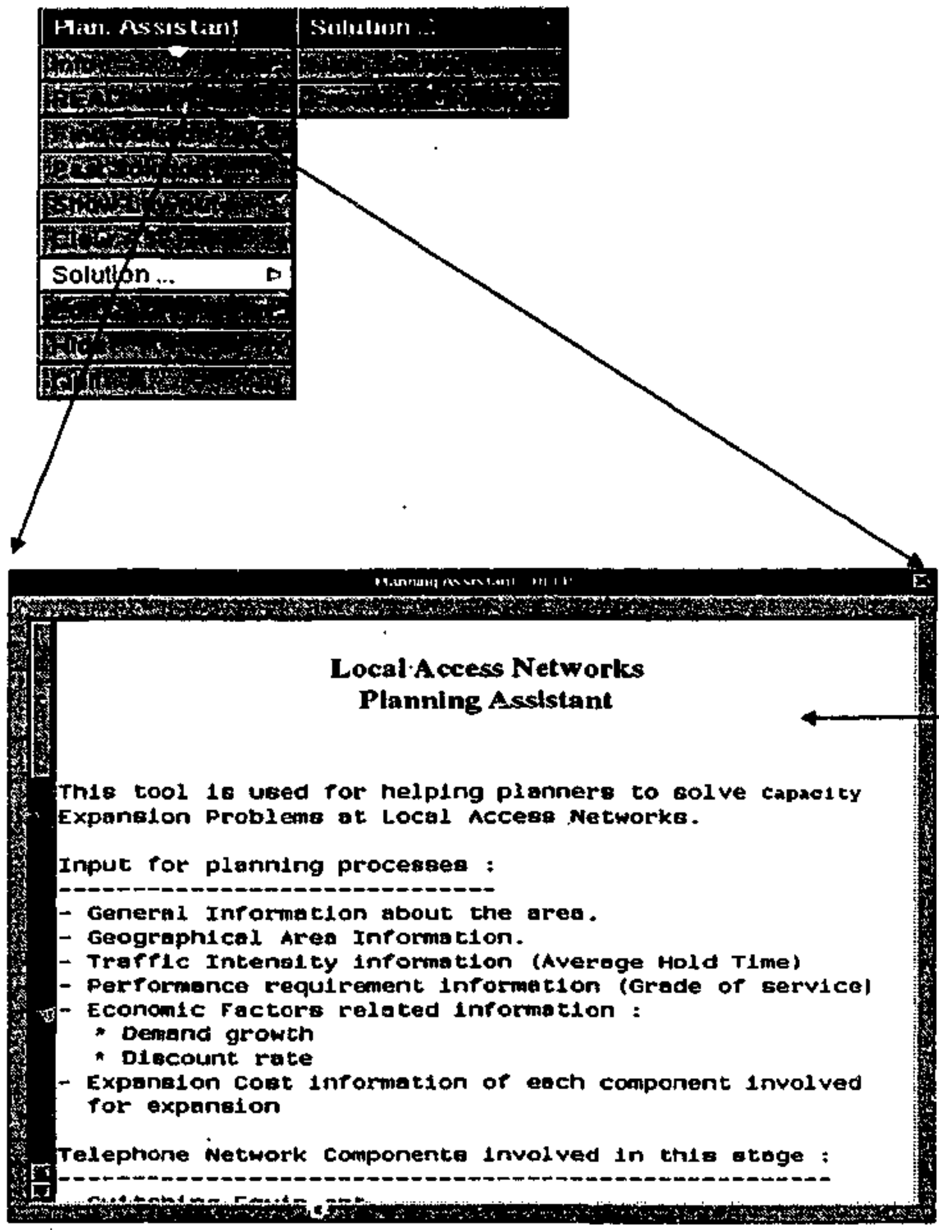
* This requirement specification data will be used to configure a network based on the selected technology; the specification is changeable according to the technology development



* Equipment required for a specific network technology

* Data of a specific product available for the network equipment group

The planning and design system also provides 'Help' information for the user. Relevant information, explanation or guidance is displayed according to the stage of the process.



* The panel provides 'help' information expected at each stage of the process

Appendix-B

Software Developed for The Prototype System

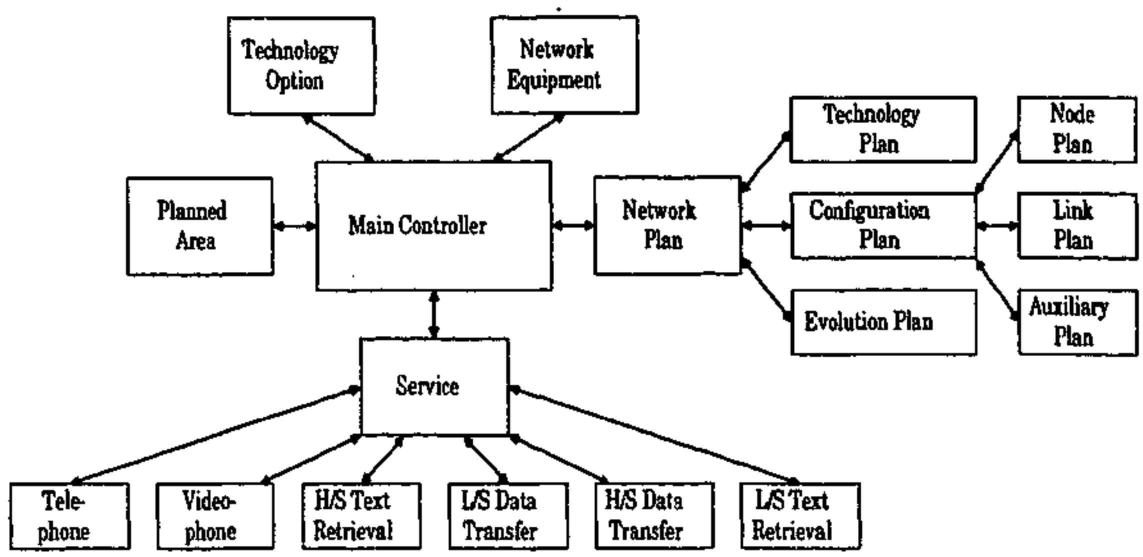
The most critical issue in the software development is choosing a set of software objects that represent the planning process, and defining the relationships between them. Once the objects and their relationships have been defined, the implementation part becomes much easier. In fact, this is one of the main advantages of using an object-oriented approach in this thesis [47-48].

In the early stage of this work, a telephone network (pure voice network) was selected as the domain application. The author expanded the scope of the domain application to the broadband network planning. This decision was made because the broadband network domain application is a more complex system that significantly involves the different types of knowledge discussed in this thesis. The broadband network is considered as a much more suitable domain for the proposed planning system.

The software implementation for the system was written by the author of this thesis using the Objective-C programming language in a NexTSTEP 3.2 environment which is based on the Mach version of the Unix Operating System [73, 75].

The planning prototype system was actually implemented as two systems, namely main and database systems. The author initially wrote a test program to verify constraint propagation by varying the input parameters to test the effect and made various modifications based on the tests. More than 10 prototype systems were developed to test the thesis.

The following figure describes a final structure of the program written for the prototype system. Rectangles represent the software objects. All of the objects shown in the figure are implemented in the main system except technology option and network equipment objects which are from the database system. They are all virtually integrated as the planning objects in the software design.



A considerable amount of code was written (more than 50,000 lines) for the prototype system. Therefore a detailed description of each program will not be provided here. However, functions implemented in the program are briefly described and they all can be categorised as follows:

I. Reading input data and propagating them to relevant objects.

II. Selecting a case with the solution from the case-base.

- Attempting a modified solution from the current solution by slightly changing the resulting solution and/or input data.
- Finding another alternative solution.
- Storing the satisfactory solution into the case-base.

III. Attempting a new solution.

- Attempting a modified solution from the current solution by slightly changing the resulting solution and/or input data.
- Finding another alternative solution.
- Storing the satisfactory solution into the case-base.

The following is a brief description of the software functions which are implemented as object methods. The list of methods and their description is divided into two parts, namely those of the main system and those of the database system. However, the list does not show all detailed functions, nor show all detailed objects.

B.1. Main system

This part includes objects which are involved in the main system. This section briefly describes the methods implemented within the following objects:

1. Main Controller.
2. Planned Area.
3. Service.
4. Network Plan.

B.1.1 Main Controller object

This object acts as a 'controller' for the planning process. It receives the input data from the user, and sends them to appropriate internal objects. It also facilitates interactions amongst all other planning objects.

<u>Main Controller object methods:</u>

- readBasicInput

This method reads all necessary input data at the beginning of the planning process.

- checkExistingNetwork

This method activates the 'readExistingInfrastructure' method when a button for checking existing network infrastructure is pressed by the user.

- checkGreenField

This method notes that the user assumes a completely new network to be planned.

- readExistingInfrastructure

This method reads all information regarding the existing network infrastructure in the area being planned.

- checkvoice

This method notes that voice traffic exists in the area being planned.

- checkdata

This method notes that data traffic exists in the area being planned.

- checkvideo

This method notes that video traffic exists in the area being planned.

- clocation

This method reads all information about the area location.

- readServicePhases

This method reads information about the services penetration phases in the area being planned.

- displayWorkingPanel

This method displays important planning details at the current state of the process. This is used by the user to analyse different options at certain stages of the process.

- thisIsACustomerArea

This method notes that an 'access area' network is being planned.

- thisIsAnInterNodalArea

This method notes that an 'internodal area' network is being planned.

- readLocationType

This method reads the type of the location.

- readAreaDiamater

This method reads the diameter of the area being planned.

- readTotalSubscribers

This method reads the total number of subscribers in the area being planned.

- lookToThePast

This method retrieves a similar case to the current planning problem, and displays the case and its solution.

- calcBandwith

This method calculates all traffic offered to the network in terms of bits-per-second.

- generatePlan

A method for triggering solution finding.

- chooseTechnology

This method triggers the process of choosing a network technology.

- imposeSelectedTechnology

A method which accepts a selected technology put forward by the user and sends it to the 'technology plan' object.

- configure

This method dimensions the current network being planned.

- reconfigure

This method repeats the process of configuring the network based on some modification imposed by the user.

- saveSolutions

This method stores a final solution as well as the planning specification details into the case-base.

- reSWCallCapacity

- gotCableSize

The two methods above retrieve specific information from the database to be used for solution-finding.

- calcInternodalTraffic

A method for calculating the bandwidth offered to the internodal network.

- dimensionInternodal

A method for dimensioning the internodal network.

- build

A method for constructing a configuration of the network using the selected technology.

- calculateCost

A method for calculating the total cost from the selected network configuration.

- showLayout

This method shows a network planning solution.

- showHelp

This method displays a 'working guide' necessary for the user when using the planning software at different stages of the interactive process.

B.1.2 Planned Area object

This object represents the entity of the area being planned. It contains the areal and other information required for the planning process.

Planned Area object methods:

- init

This method initialises the object with a default state.

- getGradeOfService

This method reads the preset grade of service that should be met in the area.

- setGradeOfService

This method sets a grade of service to be met in the area.

- otherRelevantIDs

This method takes the IDs of other relevant objects.

- getTrafficLoad

This method reads a value of the traffic load variable.

- setTrafficLoad

This method sets a new value of the traffic load variable.

- getLocationID

This method reads the value of the location ID.

- setLocationID

This method sets a new value of the location ID.

- getLocationName

This method is for reading a value of the location name.

- setLocationName

This method sets a new value of the location name.

- getStateInfo

This method is for reading a value of State name.

- setStateInfo

This method sets a new value of the State name.

- getCountryInfo

This method is for reading a value of the country name.

- setCountryInfo

This method sets a new value of the country name.

- getTypeOfArea

This method is for reading a value of the area type.

- setTypeOfArea

This method sets a new value of the area type.

- getAreaSize

This method is for reading a value of the area diameter.

- setAreaSize

This method sets a new value of the area diameter.

- getPopulation

This method is for reading a value of the area's total population.

- setPopulation

This method sets a new value of the area's total population.

- getService

This method is for reading services to be provided.

B.1.3 Service object

This object represents the network service entity. It acts as a super-object for more specific services objects.

Service object methods:

- init

This method initialises the object with a default state.

- getAveBR

This method reads the service's average bit-rate.

- setAveBR

This method set a value of the service's average bit-rate.

- getMaxBR

This method reads the service's maximum bit-rate.

- setMaxBR

This method set a value of the service's maximum bit-rate.

- getDirectionality

This method reads the service's bidirectionality coefficient.

- setDirectionality

This method sets a value of the service's bidirectionality coefficient.

B.1.3.1 HSDataTransfer

This object is a specialised object of the service object and it inherits the features of the servi object.

HSDataTransfer object methods:

- init

This method initialises the object with a default state.

- getAvePD

This method reads the value of the service's average duration in the peak hour.

- setAvePD

This method set a value of the service's average duration in the peak hour.

B.1.3.2 LSDataTransfer

This object is a specialised object of the service object and it inherits the features of the service object.

LSDataTransfer object methods:

- init

This method initialises the object with a default state.

- getAvePD

This method reads the value of the service's average duration in the peak hour.

- setAvePD

This method set a value of the service's average duration in the peak hour.

B.1.3.3 HSTextRetrieval

This object is a specialised object of the service object and it inherits the features of the service object.

HSTextRetrieval object methods:

- init

This method initialises the object with a default state.

- getAvePD

This method reads the value of the service's average duration in the peak hour.

- setAvePD

This method sets a value of the service average's duration in the peak hour.

B.1.3.4 LSTextRetrieval

This object is a specialised object of the service object and it inherits the features of the service object.

LSTextRetrieval object methods:

- init

This method initialises the object with a default state.

- getAvePD

This method reads the value of the service average duration in the peak hour.

- setAvePD

This method set a value of the service average duration in the peak hour

B.1.3.5 Telephone

This object is a specialised object of the service object and it inherits the features of the service object.

Telephone object methods:

- init

This method initialises the object with a default state.

- getCHT

This method reads the value of the service's average hold time.

- setCHT

This method sets a value of the service's average hold time.

B.1.3.6 Videophone

This object is a specialised object of the service object and it inherits the features of the service object.

Video object methods:

- init

This method initialises the object with a default state.

- getCHT

This method reads the value of the service's average hold time.

- setCHT

This method sets a value of the service's average hold time.

B.1.4 Network Plan Object

This object represent the solution entity of the planning. It is a super-object for the technology plan, configuration plan and evolution plan objects.

Network Plan object methods:

- init

This method initialises the object with a default state.

- generate;

This method generates a planning solution.

- modify

This method generate a new planning solution given changes in some of variable values typically imposed by the user.

- retrieve

This method retrieves a solution of the most similar case stored in the case-base.

- store

This method stores a satisfactory solution as well as its original planning specifications (case) into the case-base.

- reveal

This method displays a planning solution for a certain year for a particular area being planned.

B.1.4.1 TechnologyPlan

This object is a 'part of' the network plan object, and it inherits the features of the network plan object.

<u>Technology Plan object methods:</u>

- init

This method initialises the object with a default state.

- getChosenTechnology

This method reads the chosen technology for the network solution.

- setChosenTechnology

This method sets the chosen technology for the network solution.

- existTechnology

This method reads the existing network technology installed in the area.

- chosenTechnologyOverriding

This method receives the imposed network technology by the user.

B.1.4.2 ConfigurationPlan

This object is a 'part of' the network plan object, and it inherits the features of the network plan object. This object also acts as a super-object for the node plan, link plan and auxiliary plan objects.

Configuration Plan object methods:

- init

This method initialises the object with a default state.

- include

This method notes that the user would like to see the configuration plan.

- exclude

This method notes that the user would NOT like to see the configuration plan.

- getElement1

This method gets the first network element of the network solution.

- setElement1

This method sets the first network element of the network solution.

- getElement2

This method gets the second network element of the network solution.

- setElement2

This method sets the second network element of the network solution.

- getElement3

This method gets the third network element of the network solution.

- setElement3

This method sets the third network element of the network solution.

- getElement4

This method gets the fourth network element of the network solution.

- setElement4

This method sets the fourth network element of the network solution.

B.1.4.2.1 NodePlan

This object is a 'part of' the configuration plan object, and it inherits the features of the configuration plan object.

Node Plan object methods:

- init

This method initialises the object with a default state.

- getQtyN1

This method reads a value of the quantity of N1 required.

- setQtyN1

This method sets a value of N1 quantity.

- getQtyN2

This method reads a value of the quantity of N2 required.

- setQtyN2

This method sets a value of N2 quantity.

- getQtyN3

This method reads a value of the quantity of N3 required.

- setQtyN3

This method sets a value of N3 quantity.

- getQtyN4

This method reads a value of the quantity of N4 required.

- setQtyN4

This method sets a value of N4 quantity.

- calculateCost

This method calculates the cost involved of the chosen configuration.

B.1.4.2.2 LinkPlan

This object is a 'part of' the configuration plan object, and it inherits the features of the configuration plan object.

Link Plan object methods:

- init

This method initialises the object with a default state.

- getQtyE1

This method reads a value of the quantity of E1 required.

- setQtyE1

This method sets a value of E1 quantity.

- getQtyE2

This method reads a value of the quantity of E2 required.

- setQtyE2

This method sets a value of E2 quantity.

- getQtyE3

This method reads a value of the quantity of E3 required.

- setQtyE3

This method sets a value of E3 quantity.

- getQtyE4

This method reads a value of the quantity of E4 required.

- setQtyE4

This method sets a value of E4 quantity.

- calculateCost

This method calculates the cost involved of the chosen configuration

B.1.4.2. 3 AuxiliaryPlan

This object is a 'part of' the configuration plan object, and it inherits the features of the configuration plan object.

Auxiliary Plan object methods:

- init

This method initialises the object with a default state.

- getQtyA1

This method reads a value of the quantity of A1 required.

- setQtyA1

This method sets a value of A1 quantity.

- getQtyA2

This method reads a value of the quantity of A2 required.

- setQtyA2

This method sets a value of A2 quantity.

- getQtyA3

This method reads a value of the quantity of A3 required.

- setQtyA3

This method sets a value of A3 quantity.

- getQtyA4

This method reads a value of the quantity of A4 required.

- setQtyA4

This method sets a value of A4 quantity.

- calculateCost

This method calculates the cost involved of the chosen configuration.

B.1.4.3 EvolutionPlan

This object is a 'part of' the network plan object, and it inherits the features of the network plan object.

Evolution Plan object methods:

- init

This method initialises the object with a default state.

- include

This method notes that the user would like the evolution plan to be shown.

- exclude

This method notes that the user would NOT like the evolution plan to be shown.

B.2 Database system

Followings are the description of some objects from the database system.

B.2.1 Technology Option Object

This object represents the technology option as an entity. It acts as a super-object for all technology option objects used in the planning.

Technology Option object methods:

- init

This method initialises the object with a default state.

- getFeasibility

This method check whether this technology option is still feasible.

- setFeasibility

This method set the feasibility of this technology option.

- getStandardName

This method get the standard name of this technology option.

- setStandardName

This method set the standard name of this technology option.

B.2.2 Node Equipment Object

This object represents node equipment as an entity. It acts as a super-object for all objects of the node equipment used in the planning.

Node Equipment object methods:

- init

This method initialises the object with a default state.

- getGroup

This method reads the 'equipment group' to which this piece of equipment belongs to.

- setGroup

This method sets the 'equipment group' to which this piece of equipment belong to.

- getManufacturer

This method reads the manufacturer of this piece of equipment.

- setManufacturer

This method sets the manufacturer of this piece of equipment.

- getCapacity

This method reads the capacity of this piece of equipment.

- setCapacity

This method sets the capacity of this piece of equipment.

- getSpec2

This method reads the spec-2 of this piece of equipment.

- setSpec2

This method sets the spec-2 of this piece of equipment.

- getSpec3

This method reads the spec-3 of this piece of equipment.

- setSpec3

This method sets the spec-3 of this piece of equipment.

- getPrice

This method reads the price of the equipment.

- setPrice

This method sets the price of this piece of equipment.

Appendix-C

Published Papers

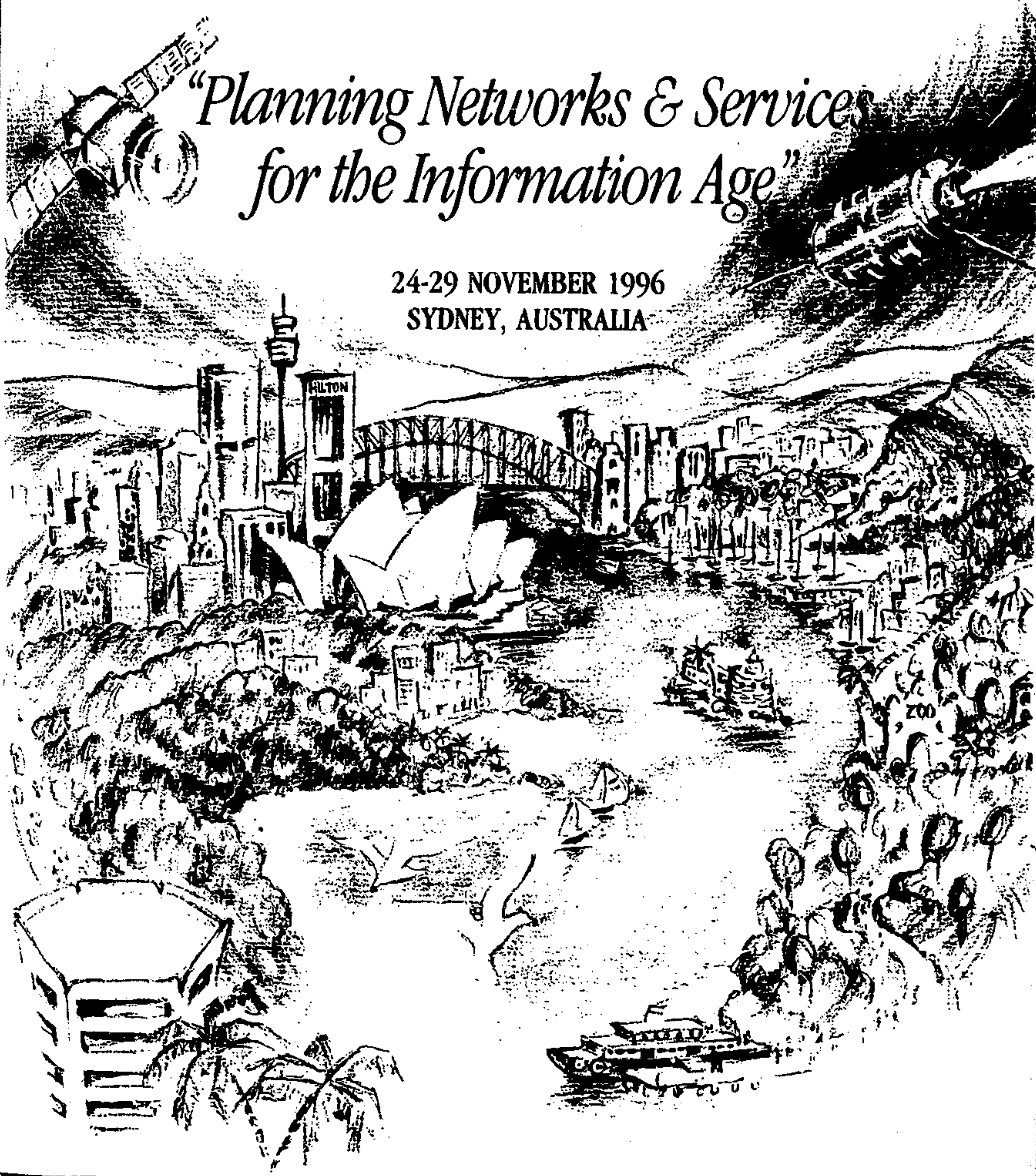
There are three papers that have been published by the author while undertaking this research work. ". All of them are attached in this appendix. Two of them were presented in the "7^{th} International Network Planning Symposium" held in Sydney, Australia, and another one was presented in the "Workshop on Intelligent Decision Support", held in Melbourne, Australia.

96

7th INTERNATIONAL NETWORK PLANNING SYMPOSIUM

"Planning Networks & Services for the Information Age"

24-29 NOVEMBER 1996
SYDNEY, AUSTRALIA



Technical Session 21: Network Planning Support

Chairperson: Prof Reg Coutts
The University of Adelaide (CTIN), Australia

Intelligent Tool for Integrated Network Planning
Afwarman Manaf, Kishor P Dabke
Monash University, Australia

Integration of the Intelligent Network Environment in General Network Planning
Beatriz Craignou, Jean-Yves David, Bernard Liau
France Telecom CNET, France

Software Support for Network Upgrade Planning
John Turner
Nortel Technology Ltd, UK

Network Planning in Telecommunications Globalization Environment
Vera D Sapozhnikova, Mai-Uyen T Nguyen
AT&T Laboratories, USA

Intelligent Tool for Integrated Network Planning

Afwarman Manaf, Kishor P. Dabke
Dept. of Electrical & Computer Systems Engineering
Monash University
Wellington Rd, Clayton 3168, Victoria, Australia
e-mail: awang.manaf@eng.monash.edu.au, kishor.dabke@monash.edu.au

Abstract: *This paper describes a tool for carrying out a multi-faceted network planning task. The task consists of several sub-tasks such as: network technology selection, traffic load determination, network dimensioning and configuration cost analysis. These tasks are dealt with by different techniques. Present planning tools are not capable of solving these tasks in an integrated way. The proposed tool is built to address the overall network planning problem and incorporates multiple techniques to solve sub-problems effectively. It uses a constraint-based planning engine as a fundamental mechanism and integrates planning rules, analytical techniques including optimization algorithms, casebase and database modules. The resulting tool is able to solve the problem in an integrated way and to increase efficiency and consistency throughout the whole network planning process.*

1. INTRODUCTION

The objective of telecommunication network planning is to find an optimal network configuration for the planned area given predicted services demand and services characteristics. The network planning tasks may include:

- network technology and architectural selection
- traffic load determination
- network dimensioning and
- configuration cost analysis.

Existing planning tools are not capable of addressing these problems in an integrated manner. Most of them only address a specific sub-problem and leave the rest to be dealt with separately. This causes inefficiency and it increases the possibility of inconsistency throughout the network planning process.

Since network planning problems are too complicated to be solved by a single problem solving technique, in practice they are solved by using many methods. The methods which are often applicable to only one particular network planning task may include optimization methods, a rules system based on experience of experts, decision making based on government/corporate policies or situational human judgement. Therefore a tool that incorporates multiple techniques is highly desirable. This research project aims to provide planners with a tool capable of helping network planners carry out the multi-faceted network planning tasks.

Many attempts have been made to realize a hybrid system using multiple techniques. There have been attempts to explore an integration of rule systems into database systems, the use of mathematical models combined with rules and a combination of rules and an object-oriented modelling [3,4,5]. Another method that has also been combined into a hybrid system is casebase reasoning [6]. The integration of methods described in previous papers seems to be ad hoc solutions. None of them was based on a unified mechanism. In contrast, this paper describes a network planning tool that uses a fundamental mechanism within a constraint-based planning engine. This mechanism allows a tight integration of planning rules, optimization algorithms, casebase and database module into a planning system. Therefore, it ensures a higher efficiency.

In this paper, the network planning problem is approached by considering it as a constraint-satisfaction problem (CSP). A solution is then selected from a choice set which does not violate a specified set of constraints. CSP can be solved through *backtrack searchs* but that can lead to what is known as an *NP-Complete Problem* which is computationally very expensive. However, methods for reducing search efforts are available. Definition and analysis of the algorithms dealing with CSP are described in [1, 2]. By using this approach, the focus is to find a *good solution* that is not necessarily the *globally optimal* one since the latter is often impossible to define precisely in practical situations.

The following section 2 describes some inter-related network planning tasks. Section 3 describes a constraint-based model as the fundamental basis of a proposed planning engine. The discussion about the planning system architecture is given in section 4. Section 5 gives conclusions.

2. NETWORK PLANNING TASKS

This section describes some network planning tasks which are presently carried out in a fragmented manner. It shows that different techniques are required to deal with each task.

2.1 Architectural and Technology Planning

Architectural planning is the most critical aspect in the network planning. It is based on uncertain information about market penetration of services (for example: probabilistic services scenarios) and a set of possible architectures with selected network technologies. The idea is to find an optimal plan for network technology deployment within a specific time span. The plan is presented in terms of a network evolution 'line' in order to anticipate predicted services demand. A more comprehensive discussion for the architectural planning can be found in Safaei and Manaf [7].

Several methods may be used to carry out an architectural planning sub-task. Search algorithms can be applied but some heuristic should be used to cope with combinatorial explosion of alternatives to reduce computational time. Some heuristic examples could be: directing a search (within *search tree*) toward forecasts that have higher probability or choosing a more conservative migration path (ie. no drastic technology change).

A probabilistic dynamic programming technique also helps to solve the problem. The decision on architectural selection is made by considering the current architecture already installed. The decision also is very much dependent on what kind of services are to be provided which are only available in forecasted forms with certain probabilities.

2.2 Network Traffic Determination

This task requires intensive computation to calculate sites' traffic load. Planners are often reluctant to analyse the results to investigate different scenarios when the appropriate computing tool is not available.

Traffic load is calculated from the given forecast of service penetration and service characteristics. Table 1 shows important characteristics of services for predicting total traffic from all services. They include maximum bit rate (B_{max}), average bit rate (B_{ave}), maximum duration of service use (D_{max}) and quality of service (eg. delay). In case of bidirectional service, (for example: video telephone) the traffic volume would be doubled because of the two ways traffic.

Service	*B_{max}*	*B_{ave}*	*D_{max}*	*Delay*
A	2M	2M	0.20s	< 80 ms
B	64k	64k	0.25s	< 10 ms
C	10M	1M	0.30s	< 10 ms
D	9.6k	64k	0.40s	< 8 ms

Table 1: Service Characteristics

An isochronous traffic is calculated in the same way as PSTN bandwidth requirements are calculated, while an asynchronous traffic uses a *bandwidth equivalent* concept. Bandwidth equivalent for each service is determined based on service characteristics and node's buffer size. The same concept can be used to determine a total bandwidth equivalent from all services. Total traffic in a planned area is the summation of all services traffic. Access network traffic is aggregated from all customers' traffic and the higher level network traffic is aggregated from those of the lower ones.

2.3 Network Dimensioning

Network dimensioning task is carried out by using some optimization methods and experience of experts. Knowledge about which network elements are available is also required.

Network dimensioning is the task of configuring an optimal network configuration in each of the planned areas. It is carried out based on a given network architecture with a selected technology and a traffic load information. Network dimensioning includes the dimension of access networks and internodal networks.

Apart from the amount of computation involved, experience of the planners is desirable to dimension networks effectively. For example, knowledge about equipment specifications is very important when selecting network elements. Planners are also forced to perform some sensitivity analysis to come up with optimized network dimensions. As an example, when the available bandwidth at any particular node is still lower than the customer bandwidth requirement in the service area, the alternative solutions are:

- to increase buffer size while considering transit delay constraint
- to increase node bandwidth by considering the backbone network and the node processing capacity
- to divide area into smaller zones with additional equipment

These alternatives must be considered in order to find an optimal solution.

Network dimensioning is not a trivial problem. In order to perform this task effectively, planners require a tool that allows them to exercise analytical models, provide them with a rules system and give them an *on-line* access to a continuously updated equipment specifications database.

2.4 Cost Analysis

The solutions of the network planning are found by taking comparison of several feasible configurations. An economic analysis is available to select the best choice. The best solution is selected from the one that gives the highest benefit based on a present worth criterion. The result of the cost analysis can be only indicative to answer the question 'How good is the configuration?'.

The economic criteria are used to decide an optimal network configuration as from among the feasible networks produced by the network planning process. They are the ultimate criteria used for network planning.

This section has described some important network planning tasks. These tasks are inter-related and need to be carried out in an integrated way. In tools available at present, the tasks are carried out in a fragmented way. There is no single system available to address all of these problems. Therefore inefficiency and inconsistencies can occur throughout the planning process.

3. CONSTRAINT-BASED PLANNING ENGINE

In order to overcome these deficiencies, we have implemented a constraint propagation engine to solve the network planning problems. The planning engine with a constraint-based model is proposed in order to build an integrated network planning system. In this section, a fundamental mechanism that underlies a planning engine is described.

In the proposed model, network planning entities are modelled as objects with their attributes and associated values. Examples of object classes are given in Table 2.

A piece of equipment, for example, has type, required quantity and capacity as its attributes. The type can be eg. a, b or c. Another example is "planned area". It has attributes of location, size, number of customers in that area and demanded services. Each attribute has its possible values.

Equipment-i:			
Type	a	b	c
Quantity	4	8	6
Capacity	50	40	10
Planned Area:			
Location	1	2	3
Size	50	70	80
# of Subscribers	8000	10000	9000
Demanded Services	k,l,m	k,l	k,l,m,n

Table 2: Objects, Attributes and Values

In this planning system, all objects and attributes are parameters for network planning and attribute values are variables. Parameters and variables become constraints in network planning. Planning constraints are propagated through the *network of relationships* of objects. Figure 1 shows some objects with some of their attributes. Attribute values of each object depend on those of other related objects.

Figure 1 illustrates how the *constraint network* represents relationships of all planning objects. Configuration cost depends on type and quantity of all pieces of equipment required. Requirements for these pieces of equipment are constrained by traffic load determination results. Results from traffic load calculation together with QoS (Quality of Service), amongst others, become constraints for dimensioning networks. At first, the input information of potential customers and services to be provided in a particular planned area are taken as constrains in the network planning process.

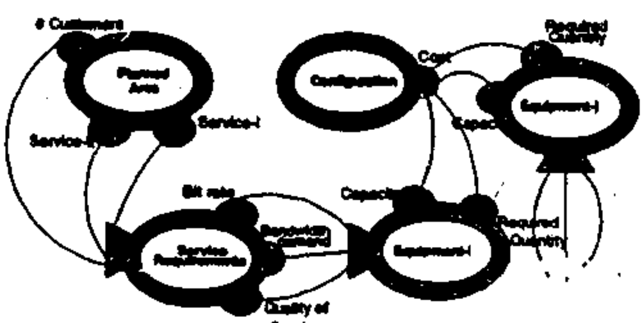


Figure 1: "Network" of Relationships

It should be noted that the constraint network represents both inter-objects relationships and relationships amongst parameters within an object. Each relationship may incorporate a simple mathematical equation or an extensive procedural algorithm. It may apply some planning rules, call specific optimization methods or involve some database retrievals.

The planning engine deduces solutions through an underlying constraint propagation mechanism. This deductive process alone is capable of finding a solution but if it is not supported by other modules it requires a high computational effort. Therefore, other modules of the planning system play their roles to restrict the solution sets that can be selected simultaneously from the choice sets. The following section describes the proposed planning system in greater detail.

4. PLANNING SYSTEM OVERVIEW

The architecture of the planning system is shown in figure 2. In this architecture, a constraint-based model is combined with planning rules, a casebase, a database and optimization methods to build an effective system to solve network planning problem efficiently.

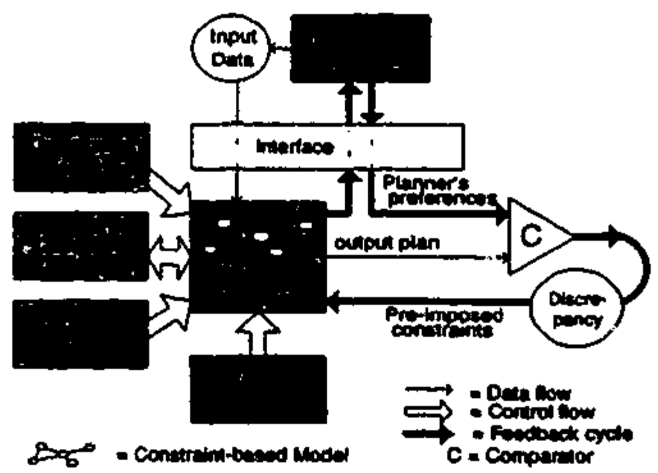


Figure 2: Planning System Architecture

The planning engine finds solutions through a constraint propagation mechanism applied to the model. The constraint-based model developed for internal representation of objects can represent a complex problem. A number of objects can be easily expanded to represent a whole network planning problem.

The planning engine also uses optimization algorithms, applies planning rules of thumb extracted from experience or expert knowledge, retrieves past cases when appropriate and queries a database to find solutions. This planning system produces a network plan from given input data such as, area information, demanded services and an existing network.

The planning tool treats input data as constraints and then find a satisfactory solution which does not violate these constraints. The solution is constructed from all objects which have attribute values consistent with all constraints within the network of objects. This solution should be optimal in some sense because it is derived from optimization procedures or planning knowledge incorporated internally. Later on, this result can be reviewed and refined through a feedback cycle with human planner intervention and acceptance.

Planning rules, casebase and database modules reduce search efforts required within planning engine. The solution is derived more quickly by using planning rules and/or using a similar case stored within the casebase. An on-line access facility to a database also enables the system to find realistic solutions and to restrict search space as well. For example, it will not explore all possible values of optimal capacity when only discrete alternatives are available when selecting a required piece of equipment.

Planning rules, casebase and database modules also provide procedural meaning to the deductive planning processes, rather than letting the engine find solutions blindly. Since casebase stores previous cases, the planners could use the tool as a training tool.

The optimization algorithms are employed to address specific well-defined problems. This approach is based on the view that there are sub-problems which are best solved by optimization algorithms. In many cases, a network plan is heavily influenced by human judgement or policies that do not necessarily conform to results produced by mathematically formulated optimization procedures. Policies and rules might have been incorporated into the knowledge base and some of them may be imposed by a human planner along the feedback cycle.

The planning engine uses pre-imposed constraints based on planner's preferences to find another satisfactory solution. This process continues until there is no discrepancy between proposed plan and the planner's preference. Through this feedback cycle, human planners still control the whole planning process and approval of the final plan. Once a satisfactory solution for a particular planning case is found, it is stored into the casebase module for future use.

The tool has some degree of intelligence because it incorporates expert knowledge used in practice by telcos. The tool also contains a casebase to store satisfactory solutions for future use. The system knowledge-base can grow to produce a better quality solution according to the lifetime of the tool. The system also improves its intelligence by expanding its casebase after solving many cases thus enhancing the quality of provided solutions.

5. CONCLUSION

The proposed system which is currently being developed has advantages over other reported systems. It is more robust because it uses multiple-knowledge sources and methods to solve an integrated planning problem and it tightly integrates planning rules, database, casebase and optimization algorithms into the constraint-based problem solver.

Since this planning tool can be used for a wide range of network planning tasks, it should help planners to carry out network planning tasks in an integrated manner. The use of the tool avoids some potential problems, such as:

- Repeatedly performing the same planning process in different planning sections of a telco
- Applying inconsistent data, knowledge or planning assumptions across different planning groups within a telco.

The tool also provides important features for performing network planning tasks, such as:

- An interactive facility to enable planners to specify preferences and to evaluate possible alternative solutions.
- An on-line access facility to geographical area data, price list, or equipment specifications.

6. ACKNOWLEDGMENT

Some of the planning knowledge incorporated in the system was kindly provided by a number of people in Telstra, Optus and PT Telkom Indonesia. The first named author was allowed to visit Optus (Sydney) and PT Telkom Indonesia in Bandung for this purpose.

REFERENCES

[1] Mackworth, A., "Consistency in Networks of Relations," *Artificial Intelligence*, vol. 8, pp. 99-118, 1977.

[2] Mackworth, A. and Freuder, E.C., "The Complexity of Some Polynomial Network Consistency Algorithm for Constraint Satisfaction Problems", *Artificial Intelligence*, vol. 25, pp. 65-74, 1985

[3] M. Stonebraker, " The Integration of Rule Systems and Database Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, no. 5, pp. 415-423,1992.

[4] G. Martini, C. Moiso, and M. Porta, "Integration of Rules and Linear Program Models in Network Traffic Management, " in *IEEE Conference on Communications*, pp. 1558-1562, Chicago, Illinois, 1992.

[5] E. Bahr, A. Holzmann, E. Mehofer and G. Pollak, " Pamela-C++: Tight Integration of Rule-based Features into an Object-Oriented Language," in J. Cuena, Ed., *Knowledge Oriented Software Design*, IFIP Transactions, 1993.

[6] D. Allemang, "Combining Casebased Reasoning and Task-Specific Architecture," *IEEE Expert*, pp. 24-33, 1994.

[7] F. Safaei and A. Manaf, "In Integrated AI/OR-Based Tool for the Planning of Broadband Multi-Service Networks", *Proc, Seventh International Network Planning Symposium*, Sydney, 24-29 November, 1996.

[8] A.M. Law, M.G. McComas, "Simulation Software for Communication Networks", *IEEE Communication Magazine*, vol. 3, March 1994, pp. 44-50.

Technical Session 5: Specialised Tools

Chairperson: Mr Jim Park
Siemens, Australia

An Integrated AI/OR-Based Tool for the Planning of Broadband Multi-Service Networks
Farzad Safaei
Telstra Research Laboratories & Monash University, Australia
Afwarman Manaf
Monash University, Australia

Optimal Design of International Communication Networks Which Exploit Time Zone Differences and Uncertainties in Traffic Loads
Les Berry, Richard Harris
Royal Melbourne Institute of Technology, Australia

DEMON: A Forecasting Tool for Demand Evaluation of Mobile Network Resources
S M Grasso, G Roso, D M Tacchino
CSELT, Italy
F M Mercuri
Telecom Italia, Italy

A Study on Common Channel Signalling Network Planning & Performance Analysis
Ock-Tae Oh, Jae-Yeon Park, Kwan-Hong Ryu
Korea Telecom, Korea
Ki-Bong Choi
Korea Advanced Institute of Science & Technology, Korea

An Integrated AI/OR-Based Tool for the Planning of Broadband Multi-Service Networks

Farzad Safaei*, Afwarman Manaf**

*Telstra Research Laboratories and Monash University
**Monash University

Abstract: *This paper deals with the planning of multi-service broadband networks. In such problems, the planner has to make high level decisions about the type of technology, the preferred topology and the nature of interactions between the various protocol layers. The current planning tools are incapable of addressing many of these issues as they are mainly focused on capacity optimisation and expansion of a fixed architecture. The paper describes a tool based on a stochastic dynamic programming model which can help with the selection of network architecture in uncertain environments. The tool is suitable for multi-service networks and can accommodate unreliable demand forecasts.*

1. INTRODUCTION

The traditional planning techniques, primarily focused on capacity optimisation and expansion, do not readily address the variety of issues facing the network designer. The increased intelligence of network elements and proliferation of technologies, protocols and management capabilities present a large number of alternative architectures to choose from. The lifetime of each technology is getting shorter which in turn increases the financial uncertainty and risks.

The planning complexity is particularly significant in the case of deploying multi-service broadband networks. In this paper, a *multi-service* network is defined as a network which can support multiple services but in terms of revenue none of those services are dominant. In other words, there is no *killer application* which by itself can justify the investment in network infrastructure and operations. The planning methodology for multi-service network has the following characteristics:

1. There is no one-to-one relationship between the network installation cost and the revenue from a particular service. By focusing on the minimisation of cost for a given service, the planner may cripple the network's ability to support others.

2. Predictions of the future demand will be more complex. There may not be enough historical data to provide a basis for extrapolation and forecast.

3. In general, the demand process will be nonstationary to cater for the emergence of new services in the future.

4. A given network architecture which has been optimised for a particular service, might be sub-optimal for others and perhaps unable to support some.

5. There may be a correlation between various services. Consequently, lack of support for some might affect the revenue from others. This could be more significant in a competitive environment where the concept of a 'one stop shop' will be attractive to customers. The network provider may be forced to support services which

could not have been justified by themselves.

Another source of complexity is a much larger repertoire of options and technologies from which the planner can select a network architecture. Choosing a particular option may 'lock' the planner into a particular path. Future deviations from this path can be quite costly. In essence, a greater number of choices will mean that the potential for inefficient planning is greater.

This type of problem will be referred to as the *architectural planning problem* or *planning problem* for short; in contrast to the *dimensioning problem* which deals with capacity optimisation and expansion of a particular architecture.

In recent years, several efforts have been made to strengthen the dimensioning techniques and expand their range to address the architectural planning concerns. Sen et al [4] develop a two-stage stochastic linear programming model for the planning of single service (leased line) and fixed architecture networks with random demand. Chang and Gavish [5, 6] look at the multi-period capacity expansion problem for a fixed architecture within a planning horizon. Lee et al [7] deal with the design of a least cost digital data service (again single service and fixed architecture). Gaivoronski [8] incorporates stochastic demands of a multi-service environment in the design of the first multiplexing stage of the access network.

2. OVERVIEW OF THE PLANNING TOOL

The tool described in this paper is designed to assist the planner in solving the architectural planning problem. The input to the tool will be a set of possible architectures (called *architectural classes*) which may be deployed either from scratch or through evolution of an existing network. Another input is a set of demand forecasts (called *service scenarios*) over the planning horizon with the corresponding measures of likelihood. Other information such as appropriate tariff levels for services, estimated cost of equipment, and operations and maintenance charges should also be provided to the tool.

The Planning Tool will then provide the planner with a 'figure of merit' for each architecture. This figure of merit is called the *Strategic Cost Metric* (SCM) and is a representation of the *worth* of a given architecture given the current estimations on future service penetrations and costs. The Strategic Cost Metric is designed to be a proper reflection of the installation cost, operations and maintenance, revenues from supported services, and the likely future costs of upgrade or growth for the given architecture.

To obtain the SCM, the Planning Tool will 'install' and dimension the architecture at the current time period according to the anticipated end-period demand (the 'periods' within the 'planning horizon' are shown in Figure 1). The installed architecture will have to face the uncertain demands of future and, depending on the particular service scenario, may go through several cycles of growth or even change of architecture. Figures 2 and 3 show the possible states that the current platform may go through in the future. The Planning Tool will optimally 'steer' the current platform in the landscape of possibilities associated with each scenario. In each time period the architecture is faced with a new set of demands as identified by the scenario and possible options for upgrade or growth with their corresponding costs and rewards. The tool will identify the cost of the optimal strategy for each scenario. Weighted average of these costs over all possible scenarios will produce the Strategic Cost Metric for the initial architecture.

As can be seen, the SCM is a reflection of combined cost of deploying a particular architecture and its future upgrade, growth, and evolution in an uncertain environment. As such, it is an unbiased indicator of the suitability of an architecture for present deployment.

The inputs and modules of the Planning Tool are described in more detail in the following Sections.

3. SERVICE SCENARIOS

Many techniques such as widespread market surveys or analytical models have been used to generate demand forecasts [9][10][11]. Naturally, these predictions may be easier or more accurate for some services compared to others. However, in general all the forecasts associated with broadband services have proved to be unreliable. For example, Hopkins et al [9] report that the number of videophones in UK by 2010 could be credibly postulated to be anywhere between 2000 and 20 million.

Here, we introduce the concept of *service scenarios*. A *service scenario* $k \in \{1,2,\cdots,K\}$ describes the *market penetration* of all services within the time horizon of study and p_k denotes the probability of scenario k being realised in the future (See Figure 1).

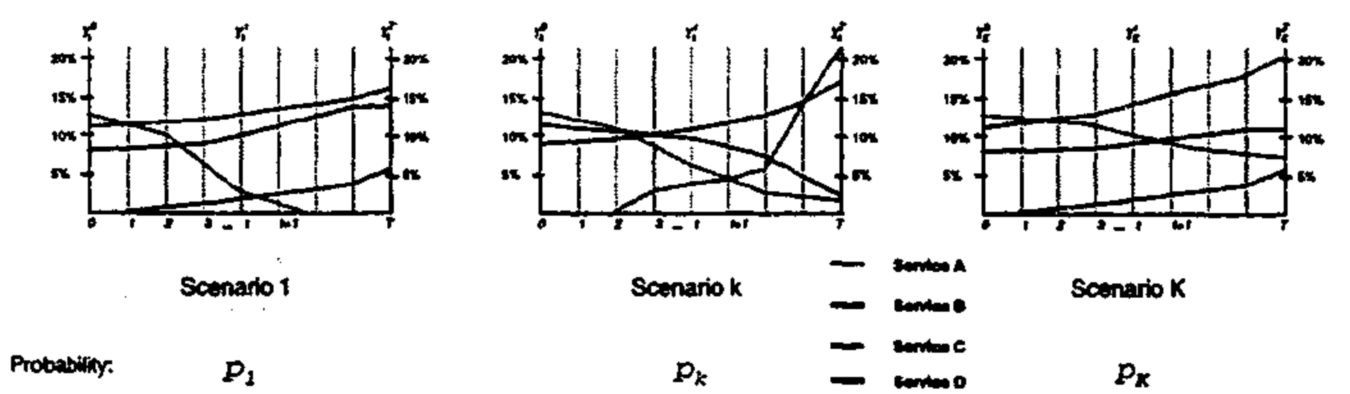


Figure 1: Service scenarios with associated probabilities

At a given time period, each service will have a particular market penetration (including zero if the service is non-existent) and is assumed to grow *linearly* to the next phase. Multiple scenarios may be based on different market surveys, modelling techniques, or adjusting the results from foreign studies.

4. ARCHITECTURAL CLASSES

By *architectural class*, we mean a broad outline of the architecture. A class would specify the topology of hierarchical levels, the types of equipment for each node and the choice of transmission, multiplexing and switching technologies. *Instantiation* of a class in a given geographical location will require making decisions about implementation details. For example, the number and placement of nodes, and the dimensions of links, ports, and switches will have to be specified following the 'blue print' of the architectural class. Naturally, to compare instantiations of different architectural classes one has to use the same geographical reach and span.

In a planning exercise, the planner will be faced with many possible technologies and deployment options. The architectural classes should be defined to capture the salient features of all these alternatives. An architectural class is an abstract entity and cannot provide any indication of the deployment cost before instantiation. However, architectural classes are useful in the study of interactions between different parts or layers of the network and the types of services supported by the network.

In [1] we have introduced several architectural classes to cover few possible ways of deploying ATM and SDH equipment in the core and regional networks. In the following, the *architectural classes* are indexed by $\alpha \in \{1,2,\cdots,G\}$.

5. DETERMINATION OF THE STRATEGIC COST METRIC

Figure 2 shows a multi-chain stochastic dynamic programming model with an initial architecture α and a time horizon T. In this Figure, each state represents a particular architecture in a given time period. The arcs indicate the cost of transitions from one state to another (not all the arcs are shown). The transition cost is equal to the discounted capital outlay for upgrade or growth plus the maintenance charge. There is also a reward associated with each state equal to the net profit.

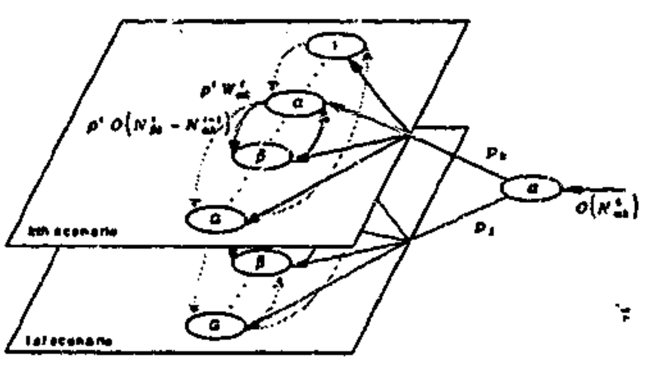


Figure 2: The structure of the architectural planning problem

Figure 3 illustrates all the possible state transitions for a particular service scenario. A *strategy* in this context can be defined as selecting a *path* in this Figure. That is, evolving the initial architecture (upgrade or growth) to meet some well-defined criteria. A strategy μ is comprised of T decisions at each time period to select the next state. For a given initial architecture α and scenario k, the *cost of strategy* μ denoted by $S^{\mu}_{\alpha k}$ can be defined as the summation of transition costs minus the rewards received in each state. The *optimal* strategy is obtained by minimising $S^{\mu}_{\alpha k}$ over all μ and is denoted by $S^{*}_{\alpha k}$. The optimal cost is derived using a procedure similar to the value iteration algorithm.

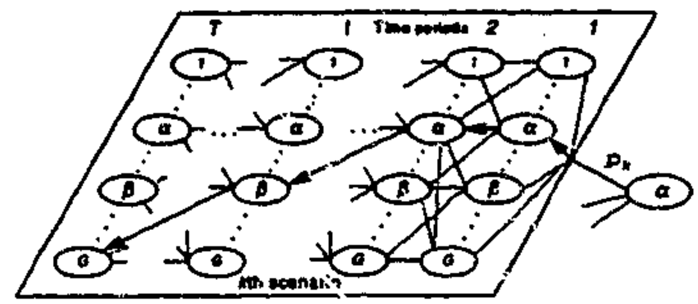


Figure 3: Network evolution for a given service scenario

When scenario k has a probability of p_k, we define the *Strategic Cost Metric*, S_{α}, for an initial architecture α as

$$S_{\alpha} = \sum_{k=1}^{K} p_k S^{*}_{\alpha k} \quad \text{or} \quad S_{\alpha} = \mathop{\mathrm{E}}_{k}\left(S^{*}_{\alpha k}\right)$$

where $\mathop{\mathrm{E}}_{k}$ is the expected value over all the scenarios. As stated before, the SCM is an indication of how costly a particular architectural class such as α will be in the long run. It represents the installation cost of the architecture to support current services. In addition, it includes the cost of later evolutions that such an architecture *on average* will endure to meet the future demands. Note that although S_{α} is in the units of dollars it does not represent an absolute dollar expenditure and is only meaningful as a comparative measure.

6. DIMENSIONING MODULE

The architectural planning problem outlined in this paper is solved using a number of sub-problems. These sub-problems are designed to be deterministic and self-contained. Each sub-problem is solved using an independent module. Computational cost of our formulation is directly related to its sub-problems which range in complexity and size. The most demanding is the *dimensioning* module. At the instantiation phase of each architectural class, this module will optimally dimension a network of class α to carry the demands of the kth scenario at the time period t while meeting the QoS and GoS objectives[1].

For the dimensioning problem, an AI tool such as that described in Manaf and Dabke [12] may be used. While that tool has wider and more general planning capabilities, its application to dimensioning problem may be seen for Figure 4. The tool can incorporate analytical models (eg. [2] and [3]), rule-based and case-based knowledge sources and input from a human operator. The human operator can override or alter the results or suggest a new approach at every stage of instantiation.

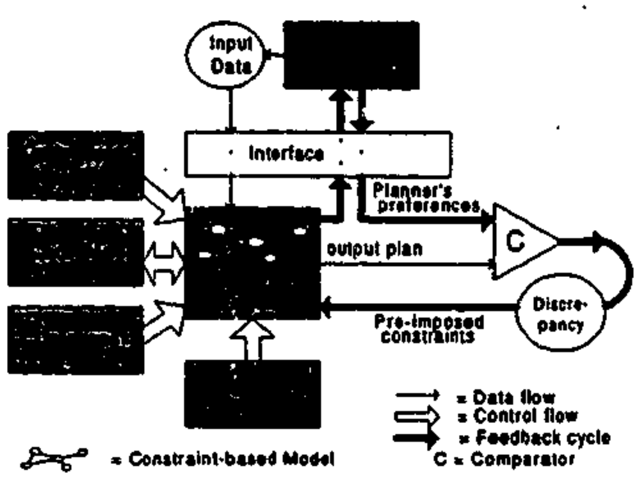


Figure 4: Dimensioning Module architecture

In this architecture, as shown in Figure 4, a constraint-based model is combined with analytical methods, rulebase, casebase and database modules to build an effective system to solve the problem. The planning engine finds solution through a constraint propagation mechanism applied to the model. The engine also uses analytical methods, applies rules of thumb, retrieves past cases stored in a casebase and queries a database to find solutions.

Rules of thumb, casebase and database reduces search space to be explored in the planning engine. The solution is derived more quickly by using rules or by retrieving a similar case from the casebase. An on-line access to database also enables the system to find a realistic solution and then restrict search space as well. For example, not all possible values of optimal capacity will be explored when only discrete alternatives are available to select a required piece of equipment.

The tool also uses pre-imposed constraints based on planner's preferences to find another satisfactory solution along a feedback cycle. Once a final solution is found, it is stored into the casebase for future use.

The outcome of the dimensioning procedure denoted by $N^{t}_{\alpha k} \in \Re^{\theta}$, is an *instantiation* of class α architecture capable of supporting the demands of the scenario within the prescribed GoS and QoS constraints (with the possibility of omission of some services). $N^{t}_{\alpha k}$ is a vector whose components $n_j : j \in \{1,2,\cdots,\theta\}$ represent the number of all the individually priced network elements required. n_j may be set to zero if the jth network element does not belong to class α.

$N^{t}_{\alpha k}$ will be used by the planning tool to calculate the cost of growth of each architecture as well as cost of migrating it to another architectural class. These figures will then be used to obtain the transition costs (identified by the arcs in Figures 2 and 3).

8. CONCLUSION

This paper describes a tool for selecting a broadband multi-service network architecture in a dynamic and uncertain

[1] It is possible that some of the services in the scenario cannot be supported over the given architecture. In this case the dimensioning routine will simply ignore these services.

world. The Planning Tool uses a stochastic dynamic programming model which deals with unreliable demand forecasts and is relatively insensitive to data errors.

Two important concepts are proposed. *Service scenarios* are convenient ways to model the demand process. *Architectural classes* are abstractions about technology and the way it is employed and are useful in the comparative study of various deployment alternatives.

The methodology presented in this paper will allow the network planner to choose among these alternatives taking into account service forecasts, evolutionary costs and the projected revenue. The resulting model is decomposed into several sub-problems which are self-contained and deterministic.

We are using this tool to investigate the introduction strategies of ATM over the existing SDH platforms.

ACKNOWLEDGMENTS

The authors wish to acknowledge Dr. Bruce Tonkin and Dr. Kishor Dabke for reviewing this paper. The permission of the Director of Telstra Research Laboratories to publish this paper is hereby acknowledged.

REFERENCES

[1] Safaei, F. et al., *"Possible Architectures for Introduction of ATM over SDH Platforms"*, Proceedings of Australian Telecommunication Networks and Applications conference, 1995.

[2] Safaei, F. and Ouveysi, L. *"Optimal Channel Assignment in SDH Networks: Large Scale Optimisation and Dynamic Analysis"*, Proceedings of 13th national ASOR conference, 1995.

[3] Safaei, F. and Ouveysi, L. *"Optimal Capacity Allocation for Support of ATM Services over SDH Unidirectional Rings"*, Proceedings of Australian Telecommunication Networks and Applications conference, 1995.

[4] Sen, S., et al., *"Network Planning with Random Demand"*, Telecommunication Systems 3, 1994.

[5] Chang, S. and Gavish, B., *"Telecommunications Network Topological Design and Capacity Expansion: Formulations and Algorithms"*, Telecommunication Systems 1, 1993.

[6] Chang, S. and Gavish, B., *"Lower Bounding Procedures for Multiperiod Telecommunications Network Expansion Problems"*, Operations Research, Vol 43, No 1, 1995

[7] Lee, Y., et al., *"Strong Formulations and Cutting Planes for Designing Digital Data Service Networks"*, Telecommunication Systems 2, 1994..

[8] Gaivoronski, A., *"Robust Access Planning for ATM Network in the Presence of Uncertain Multiservice Demand"*, Proceedings of Networks'94, 1994.

[9] Hopkins, M., et al., *"A Multi-Faceted Approach to Forecasting Broadband Demand and Traffic"*, IEEE Communications Magazine, February 1995.

[10] Wasem, O., et al., *"Forecasting Broadband Demand Between Geographic Areas"*, IEEE Communications Magazine, February 1995.

[11] Stordahl, K. and Murphy, E., *"Forecasting Long-Term Demand for Services in the Residential Market"*, IEEE Communications Magazine, February 1995.

[12] Manaf, A. and Dabke, K.P., *"Intelligent Tool for Integrated Network Planning"*, Proc. Seventh International Network Planning Symposium, Sydney, 24-29 November, 1996.

Workshop on Intelligent Decision Support

Proceedings of

The Workshop on

Intelligent Decision Support

IDS '96

Melbourne 9 September 1996

Editors:

Frada Burstein

Henry Linger

Helen Smith

A Constraint-Based System for Telecommunication Networks Planning

Afwarman Manaf and Kishor P. Dabke
Dept. of Electrical & Computer
Systems Engineering
Monash University
Wellington Rd., Clayton 3168
Victoria, Australia
e-mail: awang.manaf@eng.monash.edu.au; kishor.dabke@monash.edu.au

Abstract: This research project aims to build a planning support system that employs multiple techniques to solve planning problems. The problems can be viewed as constraint satisfaction problems. A telecommunication networks planning task is chosen as an application domain because the task consists of several sub-tasks which require different techniques to deal with. This paper describes a constraint-based planning system that tightly integrates planning rules, analytical methods, casebase and database modules to realise a more robust system. The resulting system improves the process and quality of decision making in the highly complicated telecommunication networks planning task. The human planner retains the power of choice and approval at all stages.

Keywords: constraint-satisfaction optimisation, object-oriented, telecommunication networks, planning.

1. Introduction

The objective of telecommunication network planning is to find an optimal network configuration for the planned area for a given predicted services demand and services characteristics. These network planning tasks may include:

- network technology and architectural selection,
- network dimensioning and
- cost analysis.

Since network planning problems are too complicated to be solved by a single problem solving technique, they are solved by using many methods. The methods which are often applicable to only one particular network planning task could be:

- operations research methods,
- rule-based systems,
- policy-based decision making or
- situational human judgement.

Therefore a tool that incorporates multiple techniques is highly desirable for planning telecommunication networks. This research project aims to provide network planners with a tool that *improve the process and quality of* their *decision making*.

Many attempts have been made to realise a hybrid system using multiple techniques. There have been attempts to explore an integration of rule systems into database systems, the use of mathematical models combined with rules and a combination of rules and object-oriented modelling techniques [11, 12, 13]. Another method that has also been combined into a hybrid system is case-base reasoning [14]. The integration of methods described in previous papers seem to be ad hoc solution. None of them was based on a unified mechanism. In contrast, this project aims to build a telecommunication network planning tool that uses a fundamental mechanism within a constraint-based planning engine. This mechanism allows a tight integration of planning rules, optimisation algorithms, casabas and database modules into a planning system. Therefore, it ensures a higher efficiency and consistency.

In this paper, the network planning problem is approached as a constraint satisfaction optimisation problem [2]. The optimality is determined by using some application-specific measures. This class of problems is a particular form of constraint-satisfaction problems (CSPs). One desirable feature of systems dealing with this type of problems is the ability to compare all solutions to find the optimal one.

CSP can be solved through a *backtrack search* but that leads to what is known as an *NP-Complete Problem* which is computationally very expensive. However, methods for reducing search efforts are available. Definition and analysis of the algorithms dealing with CSP are described in [5, 6]. In this project, the focus is to find a *near-optimal solution* that is not necessarily the *globally optimal* one since the latter is often impossible to define precisely in practical situations.

The following section 2 describes some inter-related network planning tasks. Constraint satisfaction optimisation is explained in section 3. Section 4 is about planning as a research topic in artificial intelligence. Models for network planning are described in section 5. Section 6 discusses a constraint-based model as the fundamental basis for a proposed planning engine. This section also gives an overview of the planning system. Section 7 contains conclusion.

2. Network Planning Tasks

This section describes some network planning tasks which are presently carried out in a fragmented manner. It should be clear that an intelligent decision support system equipped with required knowledge-base is highly desirable to carry out the task effectively.

2.1 Architectural and Technology Planning

Architectural planning is the most critical aspect of network planning. It is based on uncertain information about market penetration of services (for example: probabilistic services scenarios) and a set of possible architectures with selected network technologies. The idea is to find an optimal plan for network technology deployment within a specific time span. The plan is presented in terms of a network evolution 'line' in order to anticipate predicted services demand. A more comprehensive discussion of the architectural planning can be found in Safaei and Manaf [9].

The decision on architectural selection is heavily dictated by the current architecture already installed and it is also very much dependent on what kind of services are to be provided. Unfortunately, the services to be provided are only available in a forecasted form with a certain probability. A stochastic dynamic programming technique can be used to solve the problem. The use of AI search techniques with heuristics would also be appropriate to carry out this task. However, the architectural selection can only be done effectively by analysing site's dimensioning consequences of the options at the same time.

2.2 Network Dimensioning

Network dimensioning is the task of configuring an optimal network configuration in each of the planned areas. It is carried out based on a given network architecture with a selected technology and a traffic load information. Network dimensioning includes the dimension of access networks, internodal networks and all pieces of equipment at the node site.

Network dimensioning task can be carried out by using some optimisation methods. Dimensioning networks effectively requires not only a large amount of computation but also real world knowledge from planners. Knowledge about equipment specifications is very important when selecting network elements. In many cases, planners are also forced to perform some sensitivity analysis to come up with optimised network dimensions. As an example, when the available bandwidth at any particular node is still lower than the customer bandwidth requirement in the service area, the alternative solutions are:

- to increase buffer size while considering transit delay constraint
- to increase node bandwidth by considering the backbone network and the node processing capacity
- to divide area into smaller zones with additional equipment

These alternatives must be considered in order to find an optimal solution.

Network dimensioning is not a trivial problem. In order to perform this task effectively, planners require a system that allows them to exercise analytical models, access important rule sets and equipment specifications database.

2.3 Cost Analysis

The network plan is constructed after comparing several feasible configurations by using some economic criteria. The optimal solution can be chosen based on a present worth criterion. Cost analysis is performed for each solution produced by network dimensioning process. The result of the cost analysis can be only indicative to answer the question 'How good is the configuration?'. The economic criteria indeed play dominant role in real world network planning practices.

This section has described some important network planning tasks. At the moment, the tasks are carried out in a fragmented way. However, these tasks are inter-related and need to be carried out in an integrated way. There is no single system available to address all of these problems. Therefore inefficiency and inconsistencies can occur throughout the planning process.

3. Constraint Satisfaction Optimisation

Many problems can be viewed as Constraint Satisfaction Problems (CSPs). Planning is one amongst CSP applications. Optimisation problems studied in operations research are also constraint satisfaction problems [1]. Fundamentally, CSP is a problem constituted of a finite set of variables and a set of constraints that restricts the values the variables can concurrently take. Each variable is related to a finite domain and the variable domain is a set of all possible values that can be assigned to the variable. They could be boolean, numerical or symbolic variables. The variable-value pair is called *label* in CSP. The task is to allocate a value to each variable satisfying all the constraints. A CSP solution is then found from a choice set which does not violate a specified set of constraints.

In original constraint satisfaction problem solvers, solutions produced are all equally good. On the other hand, in applications such as telecommunication networks planning as described above, some solutions are superior to others. The assignment of different values to the same variable frequently brings different costs. The task in this problem is to find optimal solutions, where some application-specific functions are used to determine the optimality. This kind of problem is called a constraint satisfaction optimisation problem [2].

One desired feature of a system dealing with constraint satisfaction optimisation problems is the ability to compare all solutions to find the optimal one. There are at least three classes of techniques which are related to CSP problem solving namely:

- problem reduction,
- search and
- solution generation.

In CSP solving, the domains and constraints are identified in the problems and propagation of constraints enables problem reductions. There are two potential tasks involved in problem reduction:

- eliminating redundant values from the domain and variables; and
- tightening the constraints (can be based on domain-specific knowledge) so that fewer compound labels meet them resulting in smaller number of feasible solutions.

Problem reduction results in smaller domains of variables and tighter problem constraints so that there are fewer labels to consider.

Solutions of CSP can be found through a simple backtracking search. The basic procedure is to select one variable and its value at one time and to ensure a newly selected label is consistent with all the previously selected labels. If an assignment of current variable values fail, then an alternative value, when available, is selected. The problem is solved when all variable values are successfully assigned. There are two beneficial characteristics of the CSP search space, namely:

- finite search space and
- stable depth of the tree.

Additionally, search efficiency can be improved by pruning search space that contains no solution. Pruning search space really means reducing the domain size. Therefore problem reduction can be done at any stage of the search.

In the telecommunication network planning, the objective is to find an optimal solution. But it is the same as finding all solutions because all solutions is compared to find the best. However, some planning knowledge could help to find optimal solution without looking at all solutions.

4. Planning in AI

Planning is a prevalent issue in artificial intelligence and has been an active research topic since the early period of AI. Planning systems have been divided into two major classes:

- *domain-independent* and
- *domain-dependent planning system..*

A *classic* planning system reasons about the effects of actions and concentrates on identifying sequences of actions to meet certain goals when formulating a plan. A system which is built based on the classic approach falls into the domain-independent category. They use representation scheme and reasoning mechanisms that can be applied to a large diversity of problem domains [4].

Classic planning systems attempt to find all possible actions that one or more agents can execute in order to achieve specified goals given some constraints of the world in which these agents function and determine in what order these actions should take place. Process of finding a plan is called plan generation, synthesis or construction, e.g., a plan of manipulating stacking blocks by a robotic hand. However, most of these systems do not recognise complex factors external to the systems such as: government or corporate policies, stochastic information or resources constraints. Additionally, most of them deal with toy applications and their applications have not been realistic problems.

Many attempts have been made for classic systems to perform on realistic applications. Some research activities have been devoted to deal with this problem. Some of the issues that have been addressed are [3]:

- Computer Time and Resource Usage.
- Preference Constraints
- Event and Time Specifications for Actions
- Propagation of Temporal Constraints

Interested readers may consult [15] for a more comprehensive description of classical planning systems.

Problem solver of general planning systems face a very difficult problem when reasoning about the effects of actions. The core of the problem has been known as *frame* problem. The problem can be classified into of the two following combinatorial problems:

- The first: planning involves a huge number of facts that do not change when an action or event takes place. Justifications are required to confirm that these facts still hold in the world state that exists after the event. Providing these justifications entails combinatorial explosion.
- The second: the facts that do change when an action is taken depend on the circumstances in which the action is performed. These are called as *context-dependent* effects. A planning engine should encode all potential situations to make a planning system work effectively. This can also yield a combinatorial explosion.

These problems make the general problem solver computationally intractable and has been classified as *NP-hard.*

It has been generally believed that efficiency can be increased by encoding domain specific knowledge into the problem solver. Domain-dependent planning systems take domain heuristic as much as possible to encourage efficient search. In these latter systems, domain-specific knowledge plays dominant role within a planning system.

Although this paper is about a domain-dependent planning system, the proposed planning system architecture uses techniques and mechanisms which are generic in nature and can certainly be applied to other domains as well.

5. Network planning models

In this paper, telecommunication network planning entities are modelled as objects in the system and organised into classes. These models represent object attributes, methods and corresponding messages illustrating relationships amongst planning objects. Object-oriented approach is chosen because it fosters better understanding of system requirements, more systematic design and better system maintainability. Object-oriented modelling has also proved to be a very powerful tool for modelling AI systems, especially knowledge-based systems [7, 8].

This section describes a structural model of the telecommunication network planning system. It shows some objects in the system, relationships among them, and operations that characterise a network plan object class. This section also describes control and function aspects of the system [10]. The models show how a planning task is carried out by using all modules incorporated in the system.

There are, at least four other major objects involved in planning telecommunication networks, apart from the *network plan* itself. These are: planned area, network architecture, services and equipment. Network planning considers the planned area, chooses a network architecture, constructed for particular network services and specifies some pieces of equipment required. Figure 1 shows classes of planning object and some associations. Basically, a link between two classes represents bi-directional association. However, link titles indicate only one particular direction and another direction is implicit in the figure.

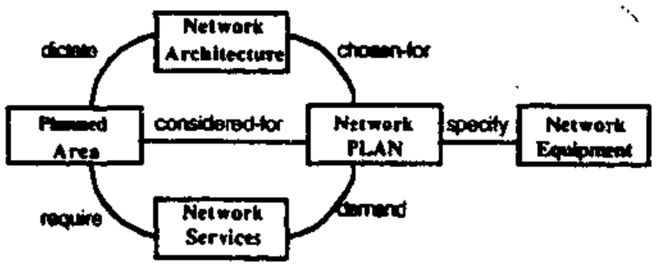


Figure 1: Classes of Planning Objects

An example of the network plan class hierarchy can be seen in figure 2. Network plan is an aggregation of nodes, links and auxiliaries. Network plan is generated for a specific time horizon (ie. year) and consists of complete specifications of nodes, links and auxiliaries. For practical purposes, a plan can be generated by including or excluding each of those components. Other valid operations for network plan include: modifying current plan, retrieving past cases (that consist of previous plans) and storing satisfactory plan into a casebase.

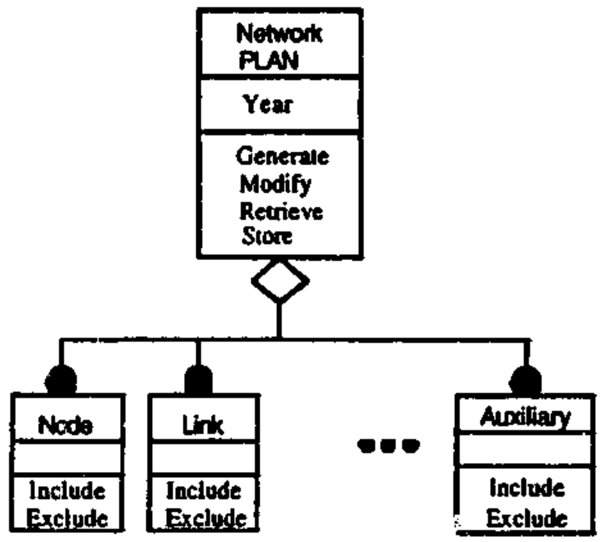


Figure 2: Network Plan Class Hierarchy

Figure 3 shows a dynamic model representing control aspects of network planning by using an *event-state* diagram. Network planning process begins with initial data input formulating the problem. A special plan should be chosen as a default plan. These input data are treated as constraints and passed to a consistency checking process for generating a plan according to the input. Alternatively, past solutions for the similar problem to the newly formulated one can be retrieved for use.

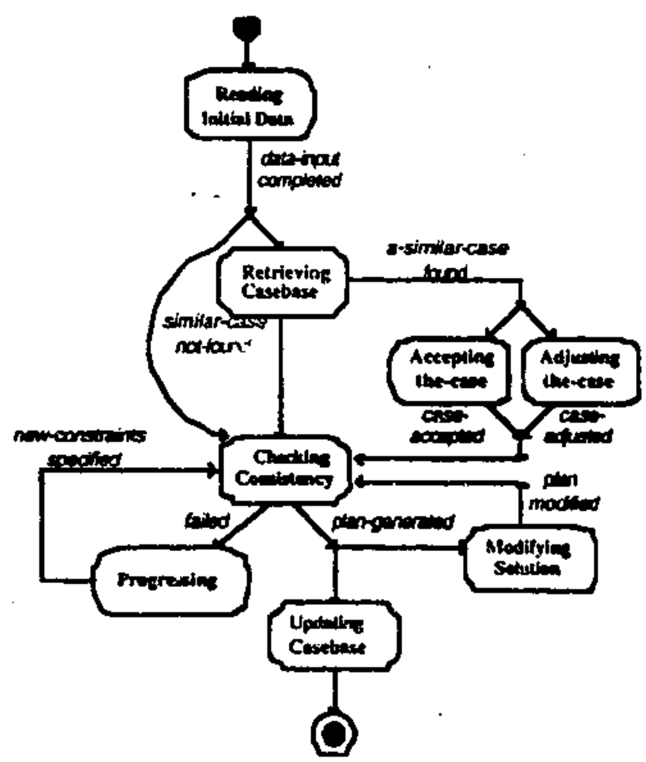


Figure 3: Network Planning Control

This previous case can be adjusted or accepted as it is to generate a solution for the new situation. It should be noted that all should go through consistency checking. The option of retrieving casebase is particularly beneficial because the past solutions might have gone through several iterations or knowledge intensive processes to become a satisfactory solution.

Once a plan is successfully generated, it can be either modified and go through *generate-and-modify* iterations or stored into casebase as a satisfactory plan. If there is no solution found, meaning that there is no plan that does not violate one or more given constraints, new specifications can be defined and *attempt-and-fail* cycle continues until there is a solution found.

The functional aspects of network planning is modelled to show computations involved in the system. A standard data flow diagram (DFD) is used for this model. Figure 4 exemplifies a higher level network planning data flow. This figure shows three major computations namely: *decide network architecture*, *calculate traffic load* and *dimension network*. It also describes functional relationships among input data (eg. aggregated demanded services), output value (eg. chosen architecture) and internal data (eg. service characteristics, architectural options and area information).

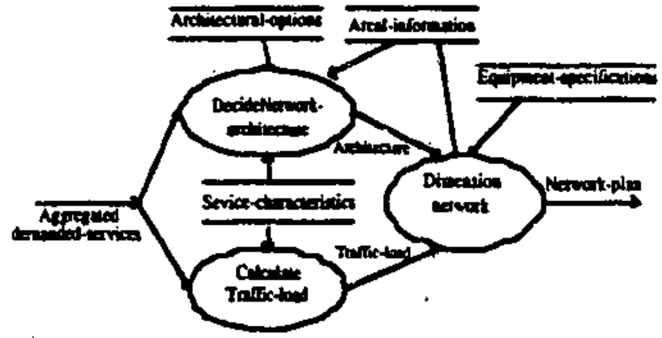


Figure 4: Network Planning Data Flow

6. Planning Support System

We have implemented a constraint propagation engine to solve the network planning problems. The planning engine with a constraint-based model is proposed in order to build an integrated network planning system. In the following sub-section, a fundamental mechanism of a planning engine is described.

6.1 Constraint-based Planning Engine

In the proposed model, network planning entities are modelled as objects with their attributes and associated values. Examples of object classes are given in Table 1.

A piece of equipment, for example, has type, required quantity and capacity as its attributes. The type can be eg. a, b or c. Another example is "planned area". It has attributes of location, size, number of customers in that area and demanded services. Each attribute has its possible values.

In this planning system, all objects and their attributes are parameters for network planning and attribute values are variables. Parameters and variables become constraints in network planning. Planning constraints are propagated through the *network of relationships*. This *dependency network* depicts relationships of all objects.

Equipment-i:			
Type	a	b	c
Quantity	4	8	6
Capacity	50	40	10
Planned Area:			
Location	1	2	3
Size	50	70	80
# of Subscribers	8000	10000	9000
Demanded Services	k,l,m	k,l	k,l,m,n

Table 1: Objects, Attributes and Values

Figure 5 illustrates possible relationships amongst planning objects. As seen in the figure, the cost incurred to a particular network plan is determined (or constrained) by all required pieces of equipment. The type and quantity of equipment needed depend on services requirements. These requirements are derived from services to be provided in the planned area and characteristics of those services. Service characteristics may include bit rate, bandwidth requirement and quality of service. Any changes made in these constraints propagate and affect other planning objects.

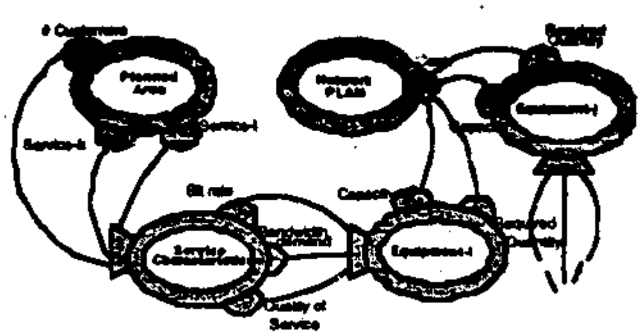


Figure 5: Objects Relationships

The constraint-based planning engine deduces solutions through an underlying constraint propagation mechanism. This mechanism is used to generate solutions. However, when it has to search exhaustively, it will requires a high computational effort. Therefore, other modules of the planning system are used to restrict the solution sets that can be selected simultaneously. The following sub-section elaborates further the proposed planning system in greater detail.

6.2 Planning System Architecture

The architecture of the planning system is shown in figure 6. In this architecture, a constraint-based model is combined with planning rules, a casebase, a database and optimisation methods to build an effective system to solve network planning problems efficiently.

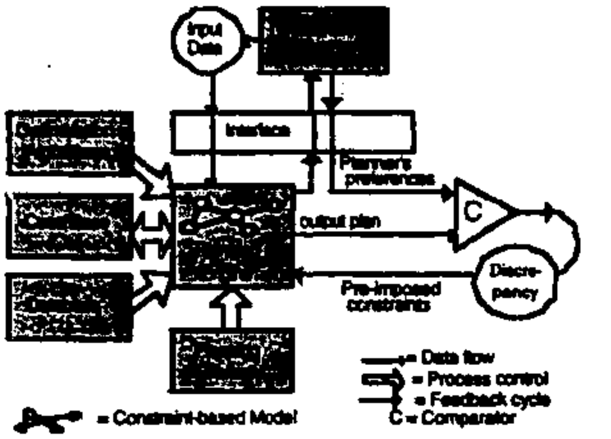


Figure 6: Planning System Architecture

The constraint-based model is developed for internal representation of objects which can represent a complex problem. This model is capable of handling a large number of objects to represent a whole network planning problem. The planning engine finds solutions through a constraint propagation mechanism applied to the model.

The planning system produces a network plan from given input data such as, area information, demanded services and existing network. To complement the deductive search, it uses optimisation algorithms, applies planning rules, retrieves past cases when appropriate and queries a database to generate solutions.

The planning tool treats input data as constraints and then finds a satisfactory solution which is consistent with these data and other constraints from *internal knowledge*. All network plan objects are established with attribute values consistent with all other constraints to construct a network plan. This solution should be optimal in some sense because it is derived from optimisation procedures or planning knowledge incorporated internally. Later on, this result can be reviewed and refined through a feedback cycle with human planner intervention.

In this system, planning rules, casebase and database modules reduce search efforts. The solution is derived more quickly by using planning rules and/or using a similar case stored within the casebase. An on-line access facility to a database would also enable the system to find realistic solutions and to restrict search space as well. For example, it will not explore all possible values of optimal capacity when only discrete alternatives are available when selecting a required piece of equipment. In addition, planning rules, casebase and database modules provide procedural meaning to the deductive planning processes, rather than letting the engine find solutions blindly.

The optimisation procedures of operations research are employed to address specific well-defined problems. This approach is based on the view that there are sub-problems which are best solved by such techniques.

6.3 The Role of Human Planner

Network planning is heavily influenced by human judgement or policies that do not necessarily conform to results yielded by mathematically formulated optimisation

6.3 The Role of Human Planner

Network planning is heavily influenced by human judgement or policies that do not necessarily conform to results yielded by mathematically formulated optimisation procedures. Policies and rules might have been incorporated into the knowledge base and some of them may be imposed by a human planner during the feedback cycle.

This system is designed as a decision support tool to allow human planners control the process and dominate in the whole decision making. Human planners control the whole planning process through the feedback cycle. Planning engine uses pre-imposed constraints based on planner's preferences to find another satisfactory solution. This process continues until there is no discrepancy between proposed plan and the planner's preference. Once a satisfactory solution for a particular planning case is found, it is stored into the casebase module for future use. Human intervention is particularly suitable because real-time response is not a critical issue in this type of planning.

The tool has some degree of intelligence because it incorporates knowledge-bases used in practice by telcos. They are implemented in the current system. The knowledge bases can grow to produce a better quality solution according to the lifetime of the tool. The tool would also improve its intelligence after solving many cases hence enhancing the quality of provided solutions. Since casebase stores previous cases, the planners could also use the tool as a training tool.

7. Conclusion

The proposed system which is currently being developed has advantages over other reported systems. It is more robust because it uses multiple-knowledge sources and methods to solve an integrated planning problem. It tightly integrates planning rules, database, case-base and optimisation algorithms into the constraint-based problem solver therefore the resulting system is highly suitable to decision support.

The constraint-based engine implemented is very effective for the planning system. It has provided a unified mechanism that enables a tighter integration of the hybrid planning system. The use of constraints derived from variety of planning knowledge in the system has also significantly pruned off search and therefore increased system efficiency. The entire planning process is under control of the human planner who has the power of approval and modification of the proposed plans.

REFERENCES

[1] Freuder, E.C., "Tools for Constraint Satisfaction," in *Proc. of the IEEE Int'l Conf. on Tools with AI*, p.7, Boston, Massachusetts, 1993.

[2] Tsang, E., Foundations of Constraint Satisfaction, Academic Press, 1993.

[3] Georgeff, M., "Planning," in Allen, J., Hendler J., Tate A.(eds.), *Readings in Planning*, Morgan Kaufmann Publisher, Inc., 1990.

[4] Wilkins, D.E., *Practical Planning: Extending The Classical AI Planning Paradigm*, Morgan Kaufman Publisher, Inc, 1988.

[5] Mackworth, A., "Consistency in Networks of Relations," *Artificial Intelligence*, vol. 8, pp. 99-118, 1977.

[6] Mackworth, A. and Freuder, E.C., "The Complexity of Some Polynomial Network Consistency Algorithm for Constraint Satisfaction Problems," *Artificial Intelligence*, vol. 25, pp. 65-74, 1985.

[7] Kaindl, H., "Comparing Object-Oriented Analysis with Knowledge Acquisition," in *OOPS Messenger*, pp 1-5, V. 5, No. 3, July, 1994

[8] Tello, E.R., *Object-oriented programming for Artificial Intelligence*, Addison-Wesley, 1989

[9] F. Safaei and A. Manaf, "Integrated AI/OR-Based Tool for the Planning of Broadband Multi-Service Networks," to appear in *Proc, Seventh International Network Planning Symposium*, Sydney, Australia, 24-29 November, 1996.

[10] Rumbaugh, J. Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., *Object-Oriented Modeling and Design*, Prentice Hall Inc., 1991.

[11] M. Stonebraker, "The Integration of Rule Systems and Database Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, no. 5, pp. 415-423,1992.

[12] G. Martini, C. Moiso, and M. Porta, "Integration of Rules and Linear Program Models in Network Traffic Management," in *IEEE Conference on Communications*, pp. 1558-1562, Chicago, Illinois, 1992.

[13] E. Bahr, A. Holzmann, E. Mehofer and G. Pollak, "Pamela-C++: Tight Integration of Rule-based Features into an Object-Oriented Language," in J. Cuena, Ed., *Knowledge Oriented Software Design*, IFIP Transactions, 1993.

[14] D. Allemang, "Combining Case-Based Reasoning and Task-Specific Architecture," *IEEE Expert*, pp. 24-33, 1994.

[15] Allen, J., Hendler J., Tate A.(eds.), *Readings in Planning*, Morgan Kaufmann Publisher, Inc., 1990.

References

[1] B. Kumar, *Broadband Communications: a professional's guide to ATM, frame relay, SMDS, SONET, and BISDN*, McGraw-Hill, Inc., New York, 1995.

[2] B. C. Lindberg, *Digital Broadband Networks & Services*, McGraw-Hill, New York, 1995.

[3] P. Darling, "Network Planning in A Changing Regulatory Environment," in *Proc. 7th Int'l Network Planning Symposium*, pp. 37-42, Sydney, 24-29 November 1996.

[4] J. Mellis, "Systematic Approaches to Network Infrastructure Planning and Performance Modelling," *B.T. Technology Journal*, vol. 14, no. 2, pp. 87-94, April 1996.

[5] K. Aly and G. Jacobsen, "Strategic Planning of Access Networks With Constraint Programming," in *Proc. 7th Int'l Network Planning Symposium*, pp. 109-114, Sydney, 24-29 November 1996.

[6] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison Wesley, Reading, Massachusetts, 1974.

[7] D. Minoli, *Broadband Network Analysis and Design*, Artech House, Norwood, Massachusetts, 1993.

[8] J. Freidenfelds, *Capacity Expansion: Analysis of Simple Models with Applications*, North Holland, New York, 1981.

[9] R. P. Davidson, *Broadband Networks: a manager's guide*, John Wiley & Sons, Inc., New York, 1996.

[10] B. G. Lee, M. Kang, and J. Lee, *Broadband Telecommunications Technology*, Artech House, Norwood, Massachusetts, 1993.

[11] K. Ward, "Network Profitability Modelling", in *Proc. 7th Int'l Network Planning Symposium*, pp. 295-300, Sydney, 24-29 November 1996.

[12] M. Capurro, R. Ravaglia, D. Giuli, "Users, Services and Traffic Modelling for Broadband Telecommunications Planning", *Electronics, Telecommunications and Transmission*, vol. 2, no. 4, pp. 433-439, Jul-Aug 1991.

[13] J. Van Duuren, P. Kastelein, F.C. Schoute, *Telecommunication Networks and Services*, Addison-Wesley, Reading, Massachusetts 1992.

[14] G. C. Kessler and D. A. Train, *Metropolitan Area Networks: concepts, standards and services*, McGraw-Hill, New York, 1992.

[15] W. Stallings, *Advances in Local and Metropolitan Area Networks*, IEEE, Inc., Los Alamitos, CA, 1994.

[16] G. C. Kessler, *ISDN: concepts, facilities, and services*, McGraw-Hill, Inc., New York, 1993.

[17] M. Fisher, T. Rea, A. Swanston, M. Wilkinson and D. Wood, "Techniques for Automated Planning of Access Networks", *B.T. Technology Journal*, vol. 14, no. 2, pp. 121-127, April 1996.

[18] E. Tsang, *Foundations of Constraint Satisfaction*, Academic Press, London, 1993.

[19] E. C. Freuder, "Tools for Constraint Satisfaction," in *Proc. of the IEEE Int'l Conf. on Tools with AI*, p.7, Boston, Massachusetts, 1993.

[20] Mackworth, and E. C. Freuder, "The Complexity of Some Polynomial Network Consistency Algorithm for Constraint Satisfaction Problems," *Artificial Intelligence*, vol. 25, pp. 65-74, 1985.

[21] Wilkins, *Practical Planning: Extending The Classical AI Planning Paradigm,* Morgan Kaufman Publisher Inc., San Mateo, California, 1988.

[22] D. Lee, K. Mehdi, J. Strand, R. Cox, and S. Chen, "Solving Large Telecommunications Network Loading Problems," *AT&T Technical Journal*, pp. 48-56, vol. 68, no. 3, May/June 1989.

[23] S. Lee, C. Wu, "A Knowledge-Based Approach to the Local Area Network Design Problem," *Journal of Applied Intelligence*, Vol. 4, pp. 7-29, 1994.

[24] J. Costa, J. Climaco and J. Craveirinha, "On the Potentialities of AI to Deal with the Combinatorial Complexity of a Telecommunication Network Planning Problem," *Engineering Applications of Artificial Intelligence*, vol. 5, no. 1, pp. 59-68, 1992.

[25] I. Fergusson, and D. Zlatin, "Knowledge Structures for Communications Networks Design and Sales," *IEEE Network*, pp. 52-58, vol. 2, no. 5, September 1988.

[26] A. Jennings, "A Learning System for Communication Network Configuration," *Engineering Applications of Artificial Intelligence*,vol. 1, pp. 151-160, September 1988.

[27] G. Martini, C. Moiso, and M. Porta, "Integration of Rules and Linear Program Models in Network Traffic Management," in *Proc. IEEE Conference on Communications*, pp. 1558-1562, Chicago, Illinois, 15-18 June, 1992.

[28] R. Davis and W. Hamscher, "Model-based Reasoning: Troubleshooting," in *Exploring Artificial Intelligence*, H.H. Shrobe and Americans Association for Artificial Intelligence (AAAI), Eds., pp. 297-346, Morgan-Kaufman, San Mateo, California, 1988.

[29] A. Manaf, and W. S. Havens, "Model-Based Diagnosis: Combining Deductive and Abductive Reasoning," *in Proc. Pacific Rim International Conference on Artificial Intelligence (PRICAI),* Seoul, Korea, September 1992.

[30] M. A. Rosenman and J.S. Gero, "The What. The How, and The Why in Design," *Applied Artificial Intelligence*, vol. 8, no. 2, pp. 199-214, 1994.

[31] A. Barr, P. R. Cohen, E. A. Feigenbaum, *The Handbook of Artificial Intelligence*, Vol. IV, Addison Wesley Inc., Reading, Massachusetts, 1989.

[32] P. Callahan, "Expert Systems for AT&T Switched Network Maintenance," *AT&T Technical Journal*, pp. 93-103, vol. 67, Issue 1, January/February 1988.

[33] M. Stonebraker, "The Integration of Rule Systems and Database Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, no. 5, pp. 415-423, 1992.

[34] K. J. Hammond, *Case-based Planning: Viewing Planning As A Memory Task*, Academic Press, Boston, Massachusetts, 1989.

[35] J. L. Kolodner, "Improving Human Decision Making through Case-based Decision Aiding," *AI Magazine*, pp. 52-68, vol. 12, no. 2, Summer 1991.

[36] M. L. Maher, and D. M. Zhang, "Cadsyn : A Case-based Design Process Model," *Artificial Intelligence for Engineering, Design And Manufacturing (AI EDAM)*, vol. 7, no. 2, pp. 97-110, 1993.

[37] A. R. Golding and P. S. Rosenbloom, "Improving Rule-based Systems through Case-based Reasoning," in *Proc. Ninth National Conference on Artificial Intelligence*, pp. 22-27, AAAI Press, Menlo Park, CA, July 14-19, 1991.

[38] H. Kaindl, "Comparing Object-Oriented Analysis with Knowledge Acquisition," *OOPS Messenger*, pp 1-5, vol. 5, no. 3, July, 1994.

[39] A. Mackworth, "Consistency in Networks of Relations," *Artificial Intelligence*, vol. 8, pp. 99-118, 1977.

[40] V. Kumar, "Algorithms for Constraint Satisfaction Problems: A Survey," *AI Magazine*, pp. 32-44, vol. 13, no. 1, Spring 1992.

[41] J. E. Doyle, "A Truth Maintenance Systsem," *Artificial Intelligence*, Vol. 12, pp. 231-272, 1979.

[42] J. de Kleer, "An Assumption-Based Truth Maintenance System," *Artificial Intelligence*, vol. 28, pp. 127-162, 1986.

[43] S. Minton, M. D. Johnston, P. Laird, "Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling," *Artificial Intelligence*, vol. 58, pp. 161-205, 1992.

[44] S. Raghunathan, "A Planning Aid: An Intelligent Modeling System for Planning Problems Based on Constraint Satisfaction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, no,. 4, pp. 317-335, August 1992.

[45] M. Jampel, "A Brief Overview of Over-Constrained Systems," in *Over-Constrained Systems,* M. Jampel, E. Freuder and M. Maher, Eds., Springer Verlag, Berlin, 1996.

[46] E. C. Freuder and R. J. Wallace, "Partial Constraint Satisfaction," in *Over-Constrained Systems,* M. Jampel, E. Freuder and M. Maher, Eds., Springer Verlag, Berlin, 1996.

[47] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*, Prentice Hall Inc., Englewood Cliffs, N. J., 1991.

[48] E. L. Cusak and E. S. Cordingley, *Object Oriented Techniques in Telecommunications*, B.T. Telecommunication Series, Chapman & Hall, London, 1995.

[49] E. R. Tello, *Object-oriented programming for Artificial Intelligence*, Addison-Wesley, Reading, Massachusetts, 1989.

[50] R. Wirfs-Brock, B. Wilkerson, L. Wiener, *Designing Object-Oriented Software, Prentice-Hall Inc.*, Englewood Cilffs, N.J., 1991.

[51] Y. Caseau, "Constraint Satisfaction with an Object-Oriented Knowledge Representation Language," *Journal of Applied Intelligence*, vol. 4, pp. 157-184, 1994.

[52] P. Coad and E. Yourdon, Object-oriented Analysis, Prentice-Hall International, Englewood Cliffs, N.J., 1991.

[53] M. Capurro, M. Coronaro and D. Giuli, "Metropolitan Area Network Planning and Design", *Electronics, Telecommunications and Transmission,* vol. 2, no. 4, pp. 423-432, Jul-Aug, 1991.

[54] M. Bruynooghe, “Backtracking,” in *Encyclopedia of Artificial Intelligence*, S.C. Shapiro, Ed., pp. 84-88, John Wiley & Sons, New York, 1992.

[55] J. C. Weber, B. K. Livezey, J. G. McGuire and R. N. Pelavin, "Spreadsheet-like Design Through Knowledge-based Tool Integration," *International Journal of Expert Systems*, vol. 5, no. 1, pp. 83-103, 1992.

[56] P. Pu, "Introduction : Issues in Case-based Design Systems," *Artificial Intelligence for Engineering, Design and Manufacturing*, vol. 7, no. 2, pp. 79-85, 1993.

[57] L. J. O'connor, M. S. Slan, D. R. Partridge and J. M. F. Lee, "A Case-based Approach to Automated Weld-process Design," *Applied Artificial Intelligence*, vol. 6, pp. 315-330, 1992.

[58] T. Bardsz, and I. Zeid, "Case-Based Reasoning for Mechanical Design," *Artificial Intelligence for Engineering, Design And Manufacturing (AI EDAM)*, vol. 7, no. 2, pp. 111-124, 1993.

[59] S. Roderman and C. Tsatsoulis, "PANDA : A Case-Based System To Aid Novice Designers," *Artificial Intelligence for Engineering, Design And Manufacturing (AI EDAM)*, vol. 7, no. 2, pp. 125-133, 1993.

[60] M. Georgeff, “Planning,” in *Readings in Planning*, Morgan Kaufmann, J. Allen, J. Hendler, A. Tate, Eds., Publisher Inc., San Mateo, California, 1990.

[61] R. Cronk, P. Callahan and L. Bernstein, "Rule-Based Expert Systems for Network Management and Operations : An Introduction", *IEEE Network*, pp. 7-21, September 1988.

[62] A. Mackworth, “Constraint Satisfaction,” in *Encyclopedia of Artificial Intelligence*, S.C. Shapiro, Ed., pp. 285-292, John Wiley & Sons, New York, 1993.

[63] R. Dechter, "Constraint Networks," in *Encyclopedia of Artificial Intelligence*, S.C. Shapiro, Ed., pp. 276-285, John Wiley & Sons, New York, 1992.

[64] J. Turner, "Software Support for Network Upgrade Planning", in *Proc. 7th Int'l Network Planning Symposium*, pp. 691-694, Sydney, 24-29 November 1996.

[65] I. Nakajima, K. Kamimura, K. Minami and H. Kakuta, "Private Network Design Method Based On AI Technology", in Proc. *Fourth International Network Planning Symposium*, Palma de Mallorca, Spain, pp. 197-202, 17-22 September 1989.

[66] N. Ford, *Expert Systems and Artificial Intelligence*, Library Association Publishing, London, 1991.

[67] J. E. Hunt, and M. H. Lee, "Towards a Knowledge-based Design Assistant," *Engineering Applications of Artificial Intelligence*, Vol. 5, No. 4, pp. 275-287, 1992.

[68] A. E. Croker and V. Dhar, "A Knowledge Representation for Constraint Satisfaction Problems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, no. 5, October 1993.

[69] A. Kershenbaum, *Telecommunications Network Design Algorithms*, McGraw-Hill, New York, 1993.

[70] F. Safaei and A. Manaf, "Integrated AI/OR-Based Tool for the Planning of Broadband Multi-Service Networks," in *Proc, Seventh International Network Planning Symposium*, Sydney, Australia, pp. 133-138, 24-29 November, 1996.

[71] L. Mason and A. Casaca, *Broadband Telecommunications*, Chapman & Hall, London, 1996.

[72] D. Allemang, "Combining Case-Based Reasoning and Task-Specific Architecture," *IEEE Expert*, pp. 24-33, vol. 9, no. 5, 1994.

[73] L. J. Pinson and R.S. Wiener, *Objective-C: Object-Oriented Programming Techniques*, Addison Wesley, Reading, Massachussetts, 1993.

[74] D. Merritt, *Building Expert Systems in Prolog*, Springer Verlag, New York, 1989.

[75] NeXT Computer Inc., *NeXTSTEP Object-oriented Programming and The Objective C Language*, Addison Wesley, Reading, Massachussetts, 1993.

[76] P.T. Telekomunikasi Indonesia, *Studies for Metropolitan Area Networks Development in Indonesia* (Internal Project Report), Bandung, Indonesia, 1996.

[77] J. L. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann Publisher, Inc., 1993.

[78] K. Ward, "Basic Network Planning", *British Telecommunications Engineering*, Issue 2, November, 1991.

[79] K. Hackbarth and F. G. Ayllon, "REFORMA: A Tool Based on PC for Telecom Network Optimization", *Electronics, Telecommunications and Transmissions*, vol. 3, no. 5, pp. 485-490, Sept-Oct 1992.

[80] W. Stallings, *ISDN and Broadband ISDN*, Macmillan, New York, 1992.

[81] D. Minoli, *Telecommunications Technology Handbook*, Artech House, Norwood, Massachusetts, 1991.

[82] J. Liebowitz, *Expert Systsem Applications to Telecommunications*, John Wiley & Sons, Inc., New York, 1988.

[83] W. Zhan, S. Thanawastien and L. Delcambre, "SIMNETMAN: An Expert System Workstation for Designing Rule-Based Network Management Systems", *IEEE Network*, pp. 35-42, September 1988.

[84] M. J. Ross, A. A. Covo, and C. D. Hart Jr, "An AI-Based Network Management System," in *Expert Systems Applications in Integrated Network Management*, Ericson et al. Eds, Artech House, Norwood, Massachusetts 1988.

[85] A. Balkany, W.P. Birmingham, and I.D. Tommelein, "An Analysis of Several Configuration Design Systems," *Artificial Intelligence for Engineering, Design And Manufacturing (AI EDAM)*, Vol. 7, No. 1, pp. 1-17, 1993.

[86] R. L. Freeman, *Telecommunication System Engineering: Second Edition*, John Wiley & Sons, New York, 1989.

[87] J. G. Pearce, *Telecommunications Switching*, Plenum Press, New York, 1981

[88] M. Green, *Knowledge Aided Design*, Academic Press, London, 1992.

[89] R. J. Horrocks and R. W. A. Scarr, *Future Trends in Telecommunications*, John Wiley & Sons, New York, 1993.

[90] R. L. Freeman, *Telecommunication System Engineering: Third Edition*, John Wiley & Sons, New York, 1996.

[91] J. Cashin, *High-Speed Networking: Technologies and Implementation*, Computer Technology Research Corp., South Carolina, 1995.

[92] I. Pita, M. de Miguel and A. Bartolome, "Planning National Telecommunication Networks," *Electrical Communication*, 2nd Quarter, pp. 188-194, 1993.