A blackboard system for active decision support in configuring telecommunication services*

Simon Parsons†§ , Tim Brown†, Simon King†† and E. H. Mamdani†

†Department of Electronic Engineering,
Queen Mary and Westfield College,
Mile End Road,
London E1 4NS, UK

Telephone: +44 71 975 5358
Facsimile: +44 81 981 0259

††Framentec-Cognitech
Department R et D
Tour Fiat, Cedex 16,
92084 Paris-La-Defence,
FRANCE

Telephone +33 1 47 96 46 00
Facsimile +33 1 47 96 46 28

ABSTRACT

As a result of the predicted growth in the telecommunications market over the next few years, it is expected that the process of configuring telecommunications services will become much more complex. As a result, the use of advanced information processing techniques to support the people who carry out configuration will become necessary. We describe the design of a prototype system that provides active decision support in such a scenario. The system combines a number of different techniques from operations research and artificial intelligence by use of a blackboard which is specially modified in order to allow the user to actively participate in the decision making process.

Keywords: decision support, blackboard architecture, telecommunications, configuration.

*This work has been supported by DESSERT, project 2021 of the CEC's RACE program. The partners in the project are Broadcom Eireann, BT, Queen Mary and Westfield College, Framentec-Cognitech, Sema Group, PTT Netherlands, Inform GmbH and Trinity College Dublin. We are particularly grateful to colleagues at BT and Broadcom for their contribution to the work described in this paper.

§Current address: Advanced Computation Laboratory, Imperial Cancer Research Fund, P.O. Box 123, Lincoln's Inn Fields, London WC2A 3PX.

1. INTRODUCTION

It is predicted that the telecommunications market will undergo great changes over the next few years [Commission of the European Communities 1991]. There are a number of reasons for this. Firstly, technological advances, such as the development of broadband networks, will permit many more services to be provided. Secondly, changes in the regulations imposed upon telecommunication service providers will result in a much more competitive market. As a result of these changes, customers will come to expect much more powerful and reliable services. It is expected that the telecommunications market of five to ten years' time will consist of many service providers, that is organisations that sell telecommunications services without necessarily providing the underlying network themselves, providing services over a multitude of networks. The environment will be highly dynamic and competitive, and the services on offer will be complex combinations of communications and data processing. One of the main ways in which service providers will be able to gain an advantage over their competitors will be to be able to provision services that both meet their customers' needs and their own long term business aims, both quickly and efficiently.

Service provisioning is the process of assessing which services should be provided to a customer, and allocating hardware and software resources to supply these services. Provisioning is currently carried out by hand [Hines and Pinnes 1988], but this is unlikely to be an efficient option in the more complex domain discussed above. As a result there is considerable interest in applying advanced information processing techniques to the problem of provisioning. However, this is not a straightforward matter. Although there are parts of the provisioning process that may be completely automated, the problem as a whole is semi-structured [Keen and Scott Morton 1978]. That is, as a result of the fact that information on how to solve the problem is incomplete, and that there is a need to deal with ill-defined concepts such as "reliable" and "best" in the search for a solution, there is no algorithm which is guaranteed to find a solution. Thus it seems that the only way to carry out provisioning efficiently is by using a decision support system [Sprague 1986] which combines the ability of a computer to solve the algorithmic part of the problem with human expertise that can deal with the non-algorithmic part.

The aim of our project is to build decision support systems to help in various stages of service provisioning, and to use the experience gained to build a toolkit from which further decision support systems may be easily constructed. We are involved in producing three prototype systems that demonstrate how decision support may be provided for the provisioning process. The first is a Customer Requirements Capture system that is used in a dialogue with the customer to help determine the services that she requires. The second decision support system is a Resource Scheduling application that allocates people and equipment to install the hardware and software necessary to provide the services. The third system is the Generation and Selection of Alternative Configurations system. This takes an abstract specification of the service required from the Customer Requirements Capture system, and decides what allocation of resources would meet the specification in a way that best suits the needs of the customer and of the service provider. The solution not only describes the telecommunications components that would need to be allocated to the customer but also any data-processing or data storage that would be required. This solution is, in turn, used by the Resource Scheduling application.

This paper describes the design of a decision support system that provides active support in the generation and selection of configurations. Using a blackboard architecture to enable flexible control, it incorporates problem solving modules that use production rules, linear programming, multi-criteria decision making, case-based reasoning, and genetic algorithms. Interaction with the user takes place through a special user layer on the blackboard that brings the user into the system as another, privileged problem solving module that not only helps to improve the current solution, but also determines which module should be invoked next.

2. OVERALL SYSTEM STRUCTURE

The Generation and Selection of Alternative Configurations application takes in a technology independent description of the services required by a customer from the Customer Requirements Capture system. This description takes the form of a list of data structures called Abstract Communication Entities (ACEs). There are different ACEs for different components of a service. For example a transport ACE details the amount of information passed between two customer sites while an encryption ACE describes the type of encoding required. Along with the list of ACEs are passed any requests the customer may make regarding the kind of premises equipment that she wants to use. The application then determines a set of suitable configurations that are compatible with both the chosen premises equipment and the set of ACEs and which thus implement the set of chosen services. If this is not possible, because for instance the customer wants to transfer data at too high a rate for her premises equipment, then those ACEs that cause the problem are identified and transmitted back to the Customer Requirements Capture application where they are used as the basis for further negotiation with the customer.

The basic architecture of the decision support system is as depicted in Figure 1. The central blackboard [Nii 1986] is a global data structure in which the current, partial, solutions are recorded. Connected to the blackboard are the various problem solving modules which, together with the user, provide the processing that is required to complete the configuration task. In a blackboard system, problem solving is an opportunistic process in which a solution is assembled piece by piece by whichever module can best contribute at the time. Each module, or knowledge source in blackboard parlance, monitors the blackboard for partial solutions to which it can contribute. When it can work on a particular partial solution, it creates a knowledge source activation record (KSAR) which it passes to the control module. The control module decides which problem solving module is to be run, based on knowledge about the module and the nature of the solution on which it wants to work. The blackboard is updated by the module in question, and the cycle begins again.

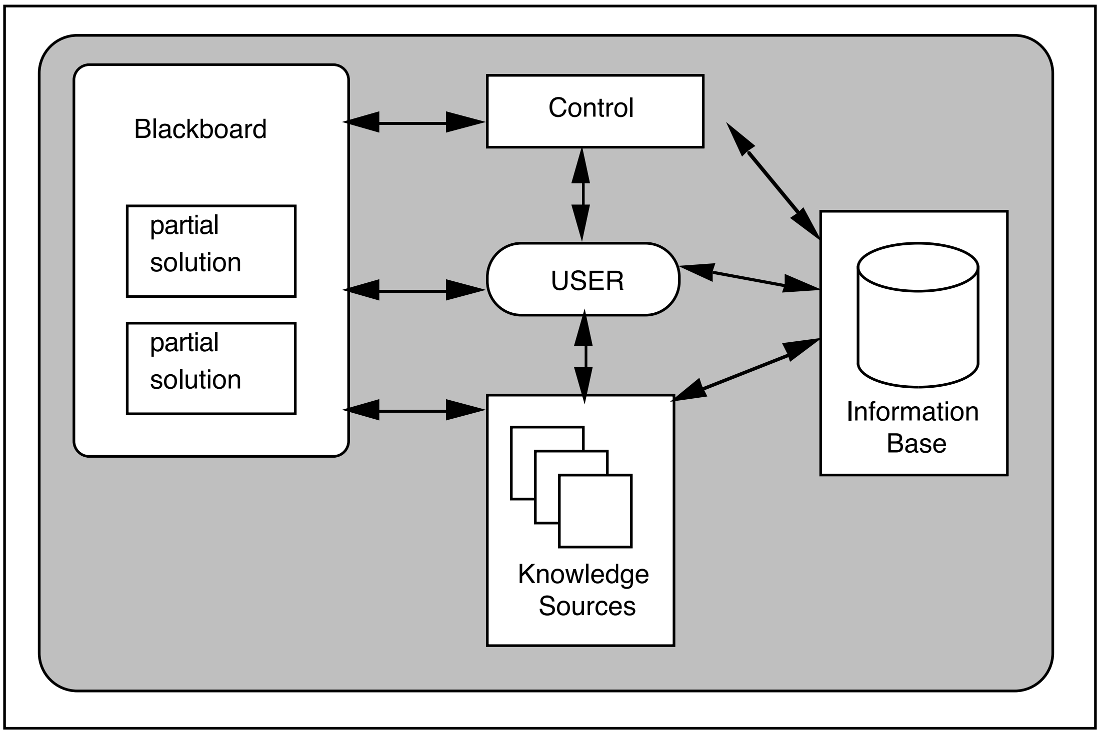

Figure 1. The architecture of the Generation and Selection of Alternative Configurations Application

Different problem solving modules can complement each other. For example in the Generation and Selection of Alternative Configurations system there are a number of evaluation functions which determine how reliable, costly and secure a configuration is. These are complemented by a multi-criteria decision making tool which takes their output and combines it into a single measure for comparison with other configurations. The modules also provide competing methods of problem solving. For instance, there is a tool that uses linear programming to determine how many of each type of ISDN primary rate access channels are required at a particular customer sites. This competes with a simple rule-based tool that does the same job, but which will be appropriate in different situations. For instance, at an early stage in the configuration the rule-based tool may be used to give a rough answer, which may be changed as a result of the output of other tools. Later on, when the configuration is being finalised, the linear programming tool might be used to give an optimal solution.

Using such complementary and competing tools provides a very flexible problem solving environment, and is a typical feature of blackboard systems. Another advantage of using a blackboard is that it provides a simple means of integrating different problem solving techniques. Using a blackboard enables each technique to be written as a separate knowledge source, using its own data structures. This avoids the “shoehorning” of one method into another which is a common result of combining different techniques, for instance writing cases as rules, or rules as cases, to combine rule and case-based reasoning [Dutta and Bonissone 1990] [Golding and Rosenbloom 1991].

3. THE STRUCTURE OF THE BLACKBOARD

The blackboard is divided into layers in order to improve efficiency [Nii 1986]. Each layer contains solutions that are built up from a particular type of data structure, and is named for that type of data structure. The mappings between the layers, and hence the data structures, are carried out by the knowledge sources. Each knowledge source only searches the layer that contains data on which it can operate, and so is associated with a particular layer. The structure of the blackboard and the arrangement of some of the knowledge sources around it is given in Figure 2. The knowledge sources are indicated by annotated arrows that show the transformations that they perform.

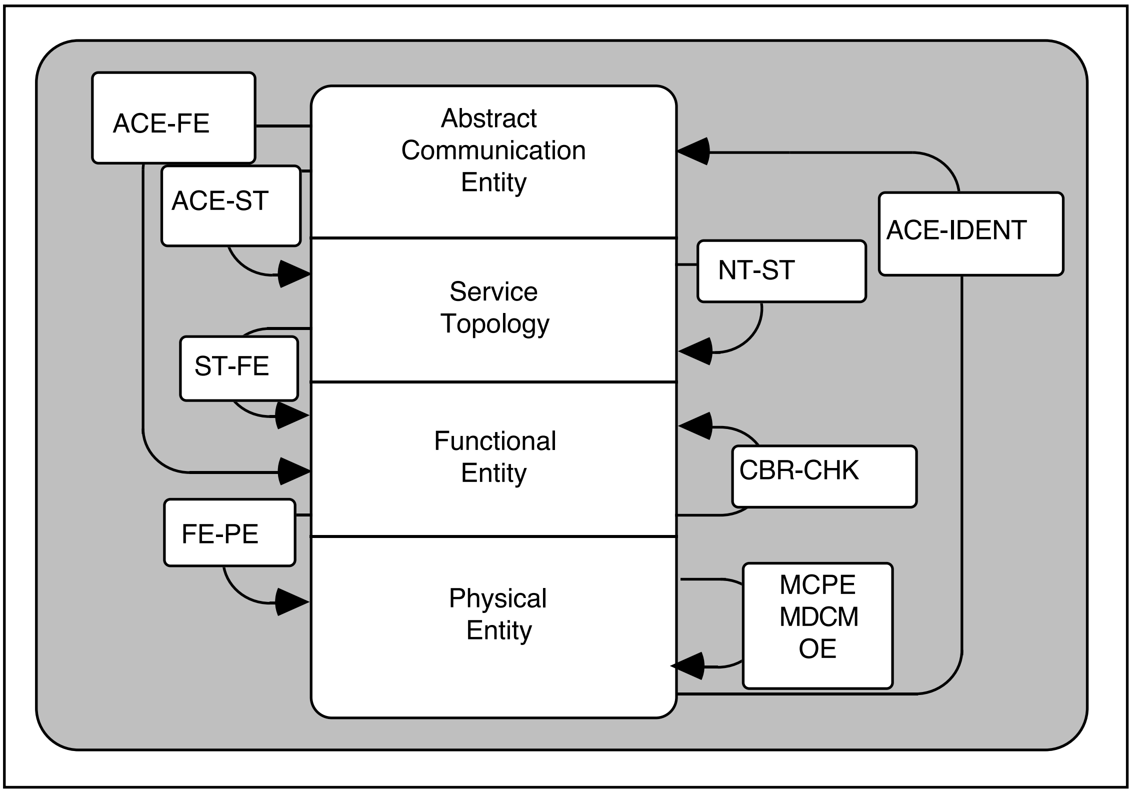

Figure 2. The structure of the blackboard and the arrangement of the knowledge sources

The problem arrives from the Customer Requirements Capture in terms of sets of ACEs that are placed on the ACE layer at the top of the blackboard. Each set of ACEs is an abstract description of a particular service requested by a customer. All the sets of ACEs for a particular customer are mapped onto a single Service Topology (ST) on the next level down. This gives the connectivity of the access links between customer sites and the trunk network, and provides a structure on which the requirements from the different sets of ACEs may be accumulated. The Service Topology then maps to a number of descriptions in terms of Functional Entities, on the FE layer, which describe, for instance, that a particular connection will be a primary rate ISDN link, without saying exactly which components will be used. When the FEs are mapped down to physical entities (PEs) such decisions are made, and the exact physical components that will be allocated to the customer are identified.

Each layer on our blackboard is implemented a separate process. On each cycle, the conditions of every knowledge sources are evaluated against the state of the blackboard, and a separate agenda drawn up for every layer. The order of KSARs on this agenda reflects the priorities of those knowledge sources which□are determined by the system designers. Each layer then runs the knowledge source corresponding to the highest priority KSAR and erases the agenda from which this KSAR was selected, preventing knowledge sources being triggered when there is no partial solution for them to work on. The cycle then begins anew.

4. KNOWLEDGE SOURCES

The knowledge sources, which transform partial solutions and map between layers on the blackboard, form the heart of the Generation and Selection of Alternative Configurations system. The knowledge sources required by the application can be classified into a number of different classes. The basic kind of knowledge source is a **down-mapper** which progresses a partial solution down the blackboard. Most of these knowledge sources move a solution down one level at a time, though there is an exception:

ACE-ST: this knowledge source uses data about the transport ACE to create the basic connectivity pattern of the service topology.
ST-FE: this knowledge source provides the mapping from the complete service topology to a set of functional entities and may generate more than one possible set of such entities. For instance in some circumstances the requisite service may be implemented over more than one kind of network.
FE-PE: this maps from functional entities to physical entities, largely on the basis of what equipment is already present in the access network since our aim in provisioning is to allocate existing equipment rather than arrange for the installation of new equipment.
ACE-FE: this uses case-based reasoning to provide a direct mapping from the list of ACEs to a list of functional entities. The mapping is based on cases drawn from previous experience, and the knowledge source includes a limited ability to modify the cases to fit the current situation.

The other major type of knowledge source is the **value-adder** which operates on attribute values at a particular level of the blackboard to extend the solution:

NT-ST: this knowledge source fleshes out the basic service topology using information from non-transport ACEs such as those that represent encryption.
MCDM: a multi-criteria decision making knowledge source, based on fuzzy weighted averaging, helps the operator assess which possible configuration best serves the interests of the customer and service provider.

This makes its assessment based upon information from the:

OE: objective evaluation knowledge sources which range from rule-based implementations of simple heuristics to establish the degree of security of a solution to complex cost estimation functions based on operations research methods.

We also have a **backtracker** knowledge source which unwinds failed solutions up the blackboard:

MCPE: if the suggested configuration is incompatible with a customer's requested premises equipment, it may be necessary to reassess the functional entities by backtracking to this level.

and **consistency-checkers** which monitor the consistency of solutions:

CBR-CHK: this checks the output of the case-based reasoner and patches the solutions where possible by making heuristic changes to the list of FEs.
CPE-CHK: this operates at the PE level and compares the requirements placed on the customer premises equipment with the capabilities of that equipment.

as well as an **up-mapper** Knowledge Source to allow data at a higher level of the blackboard to be established from data at a lower level:

ACE-IDENT: In the event that it proves impossible to satisfy the customer's demand for services given her choice of premises equipment, this knowledge source identifies those ACEs that have caused the problem.

There are other, more specialised, control knowledge sources whose purpose is to circumvent particular problems or to guide the system to particularly efficient solutions. These need not concern us here.

5. MIXED INITIATIVE PROBLEM SOLVING

Since the systems that we are developing are decision support systems it was imperative that we made the Generation and Selection of Alternative Configurations system as interactive as possible. As a result we found ourselves working to some extent against the blackboard system approach in that blackboards are usually allowed to run under their own guidance with little, if any, interference from a human operator. To ensure that we had a system that interacted with the user in a suitable way it was decided to make the system reactive; that is we tried to ensure that when the operator inputs something to the system the latter considers the new input immediately. Two methods were adopted in order to create this kind of reactivity. Firstly a user level was introduced. This is a special layer on the blackboard through which the user can interact with the system. Secondly the knowledge sources were made as small as possible. These ideas are discussed further in the following sections.

5.1 The user level

The first function of the user level is to provide a window onto the blackboard. Through this window the user can read information from every layer of the blackboard in an easily comprehensible form. The user may also write to any layer on the blackboard through the user level. The user layer has the highest process priority so that the knowledge sources at this level run as soon as the new data is detected, placing this data at the appropriate level of the blackboard. When the user posts information in this way it has a much higher priority than information posted by any knowledge source, and is thus acted upon as soon as possible. In this mode the user level acts as a communication server. The user level also provides a means for the user to observe and alter the control regime of the blackboard. Through the user level the operator may question the system as to why certain knowledge sources were put to the top of the agenda, or why a particular KSAR was selected for execution. The operator may also wish to disable the execution of knowledge sources so that particular lines of enquiry are pursued as quickly as possible in line with her intuition about the solution. Furthermore the operator may want to alter the triggering conditions of particular knowledge sources so that they activate under different conditions, thus dynamically tailoring them to a particular problem. Thus the user level is a way of making the user more in control of the system than is usually the case with blackboards systems. It must, however, be noted that the user must be careful when exercising this control.

5.2 The granularity of the knowledge sources

The innovation here was that the knowledge sources were made as small as possible. The rationale behind this decision was the following. Knowledge sources represent a number of operations that transform the partial solutions on the blackboard. There is no mechanism in a typical blackboard system for interrupting a knowledge source while it is executing, indeed the whole idea of a blackboard is to give each knowledge source complete control while it is doing whatever it does. If you interrupt a knowledge source you may get a partially transformed partial solution which will be unable to trigger any further knowledge source, unless of course the knowledge source is an anytime algorithm [Dean and Boddy 1989], and will thus be left to fester on the blackboard. Thus it is not a reasonable to look to introduce reactivity by allowing the user to interrupt the blackboard. The only possible approach, therefore, to increasing the speed of the system's response to the user is to reduce the time that the user has to wait for the knowledge source that is executing when the new input is made to complete its computation. This seems feasible since it may well be possible to break the necessary pieces of functionality down into atomic operations each of which could be a knowledge source in its own right.

For instance in the Generation and Selection of Alternative Configurations one obvious knowledge source is that which maps ACEs down into Service Topologies. This could be viewed as a single piece of functionality with, say, a large number of forward chaining rules that aggregate the effect of all the sets of ACEs that support all the services being supplied to the customer and generates a partial solution which indicates all the connections necessary, the bandwidth of each connection, and additional services such as encryption, security and help desk facilities. Such a knowledge source would be quite complex and take a considerable time to execute. It could be replaced by a number of knowledge sources, each of which worked on a particular type of ACE. Thus a transport-ACE-ST knowledge source would establish the connectivity required by the customer, and instantiate the resultant service topology with the necessary bandwidths. This partial solution could then be augmented by a second knowledge source that maps from data processing ACEs, such as those that provide encryption, to features of the service topology such as processing at the customer's or service provider's premises. These ACEs would be smaller and therefore faster executing allowing the system to scan its agenda more often and thus allowing the user a more interactive role.

6. WORKED EXAMPLE

A newspaper wishes to subscribe to a set of services linking two journalist, one editor and a printer. Representatives from the newspaper and Service Provider organisations meet to discuss the newspaper's communication needs, and together with the help of the Customer Requirements Capture (CRC) application they decide on a set of two services. The CRC application maps the chosen services to two sets of Abstract Communication Entities (ACEs), illustrated in figure 3, and presents these as input to the Generation and Selection of Alternative Configurations (GSAC) application. Each set of ACEs represents the information to describe a single service in a technical yet implementation independent manner.

Service 1 allows articles to be transferred between the two journalists and to and from the editor at a medium fast rate. The articles sent are also stored on a database managed by the Service Provider. Documents stored in this way may be retrieved by any of the parties participating in that service. The Data Link ACE describes the type of traffic required to implement this service (e.g.

bandwidth) and the associated usage profile captures expected volume of traffic between each pair of sites at each hour of the day. The Back Up ACE summarises information such as the type of data to be stored and the desired speed of retrieval. The usage profile associated with the Back Up ACE describes the expected rate of retrievals by each of the participating sites.

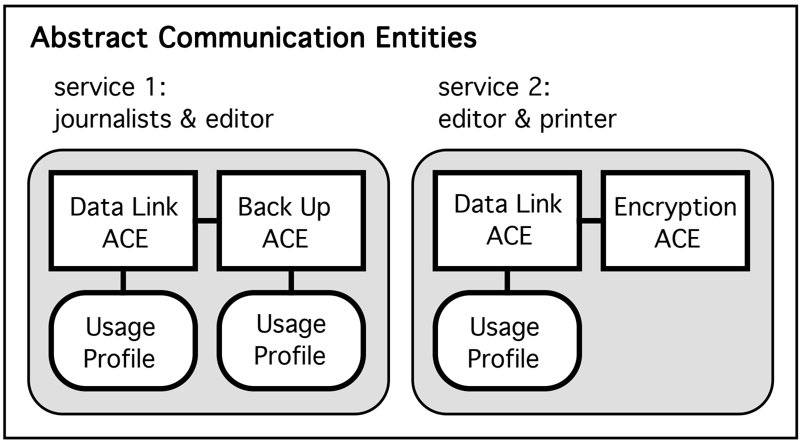

Figure 3. Services represented as Abstract Communication Entities

Service 2 is used to send complete versions of the newspaper from the editor to the printer, and is implemented by a very fast one way link. The service is high security and these requirements are captured by the parameters of the Encryption ACE.

6.1 Service Topology Level

Knowledge sources work on information represented at ACE level to compile a single topology object on the Service Topology level of the blackboard. This object, illustrated in figure 4, represents the required connectivity of the parties involved. Superimposed upon this is information about the traffic associated with each link, and compatibility constraints between various sites and access networks.

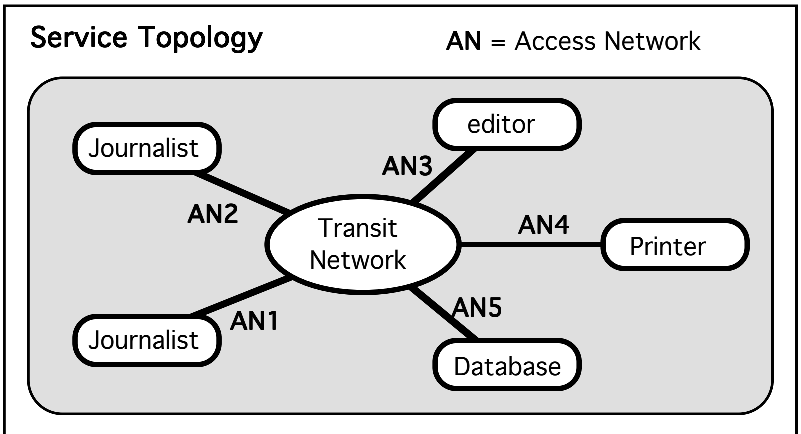

Figure 4. Services represented as a Service Topology

For each access network one or more possible transit network connection points are identified, except for AN5 which connects the database. At this stage the choice of which of the many possible Service Provider's databases to use has not been decided, the implications of that choice are explored as the problem is progressed to the other black-board layers.

6.2 Functional Entity Level

The process of mapping the problem from the Abstract Communication Entities layer to the Service Topology layer involves little overt decision making. In contrast moving the problem to the Functional Entity layer involves tackling the main questions surrounding configuration, which are in this case:

- Which type of network to use for each service on each link?
  This is currently a choice between Integrated Services Digital Networks (ISDN), Public Packet Switched Data Networks (PPSDN), and Metropolitan Area Networks (MAN).

- Which database to use?
  The Service Provider has several databases, some of which are infeasible for this problem because they are not suitable for holding mixed text and graphics data or they do not support the customer's choice of query language.

- Where should documents be duplicated?
  When articles are to be both saved in the database as well as sent directly to another party then the document needs to be duplicated at some place. This could be done at the sender's site, one copy is sent to the database and the other to the nominal receiver, or alternatively the receiver duplicates the document and forwards a copy to the database. Other possibilities are for documents to be sent via the database or for duplication to take place at an access network to transit network interface. Duplication is an automated process requiring the provisioning of a duplicating entity at appropriate locations.

The answers to each of these questions is dependent on the choices made when answering the others. Even the choice of network type for AN4, which links the Printer, relates indirectly to the choice of the database and duplication policy even though the Printer does not particapate in the service involving database backup. The system proceeds by a best first depth strategy. Heuristics embodied in the ST-FE knowledge sources identify the most critical decisions and start to instantiate answers. In our example the Editor's access network is identified as a pivotal element since the Editor is the only party participating in both services. The need for a fast, high capacity link to the Printer suggests the choice of a Metropolitan Area Network whilst the need for flexible communications with the Journalists suggest an Integrated Services Digital Network. The Editor's site can, of course, be allotted the equipment necessary to communicate over both network types, but this increases expenditure on customer premises equipment.

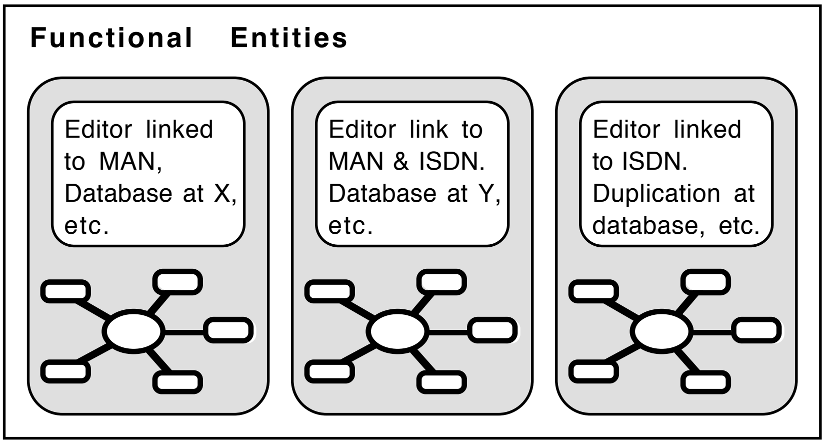

Figure 5. Services represented as Functional Entities.

The user directs the system to create several complete configurations where the critical questions are answered in different ways. Those questions that are considered less important are answered according to the heuristics embedded in the knowledge sources. This gives rise to several objects on the blackboard, each is based on the Service Topology representation but has different Functional Entity instantiations according to the answers given to the 'critical' questions. Each object represents an alternative configuration and is labelled with the assumptions on which it is based. Figure 5 illustrates three such objects in practice there would be about ten to fifty alternatives at the Functional Entity level.

6.3 Physical Entity Level

Two main tasks occur at the Physical Entity blackboard layer. Firstly, Functional Entities are mapped to Physical Entities and secondly the resulting configurations are evaluated. Physical Entities are objects such as a hardware encryption board or a connection to a particular switch at the access network to transit network interface. Broadly speaking the choices made at this level have only relatively local effects, which is to say that most mappings can be performed separately according to criteria such as compatibility with existing customer premises equipment, or minimisation of costs to the Service Provider.

Once a configuration is complete Objective Evaluation knowledge sources come in to play. Each OE knowledge source is customised to evaluate a particular part of the configuration according to it's own criterion. For example there is a Transit Network Flexibility Evaluator, an Access Network Reliability Evaluator, a Customer Premises Maintenance Cost Evaluator, and so on. Figure 6 illustrates the state of the Physical Entity blackboard layer; each configuration now consists of representations of physical rather than functional objects, they are still labelled with the assumptions on which they were based, but are now also linked with their evaluations.

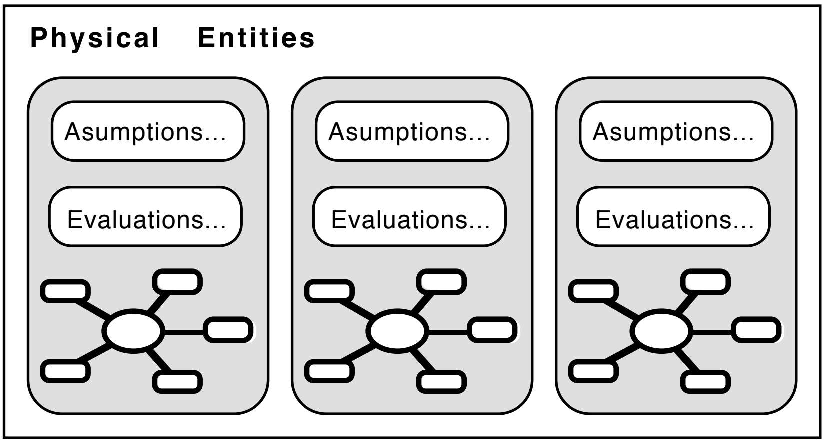

Figure 6. Services represented as Physical Entities.

The evaluations associated with a configuration are aggregated into a single measure of that configurations quality in order that all the configurations can be compared. Aggregation is based on the user supplying weights for the relative importance of the evaluation criteria. This allows the trade-offs associated with the earlier assumptions to become apparent. The user can then initiate backtracking to take advantage of this knowledge in order to improve on the current best configuration.

Once a configuration is found which the user is satisfied with the processing of the problem is complete as far as the Generation and Selection of Alternative Configurations application is concerned. The Physical Entity representation of the best configuration is passed to the Resource Scheduling application in order to devise a work programme to implement the configuration.

7. CONCLUSION

This paper has described the design of a system for the generation and selection of alternative configurations of telecommunications services. The system is currently being implemented, and will be demonstrated in September 1993. The central idea of the systems is that machine and user together form a team, with the system providing the kind of active decision support advocated by Raghavan [1991] and Dolk and Kridel [1991]. Thus the machine does not passively sit back waiting for the user to ask it how a particular configuration would rate, or whether two particular components may be connected to one another. Instead, as soon as the service request is posted to it the system sets to work, providing a range of possible configurations whose details will depend upon the modules that were used to generate them. The user can sit back and watch this happen, waiting for the system to generate a complete solution that she can then rate using the MCDM tool, or she can get involved in the generation of a solution by making modifications to the partial solutions as they sit on the blackboard. If the user intervenes in this way, the system will then use its compiled expertise to work on the user-modified partial solutions. This mode of working is thus completely flexible allowing the users skills to complement the ability of the machine and providing powerful support for decision making in this complex domain.

REFERENCES

Borch, O. J. and Hartvigsen, G. (1991) Knowledge-based systems for strategic market planning in small firms, Decision Support Systems 7, pp 145–157.

Commission of the European Communities (1991) Perspectives for Advanced Communication in Europe: 1992, Impact Assessment and Forecasts Volume II, CEC, Office for Official Publications of the European Communities, Luxembourg.

Dean, T. and Boddy, M. (1988) An analysis of time-dependent planning, Proceedings of the 7th National Conference on Artificial Intelligence, Saint Paul, Minnesota.

Dolk, D. R. and Kridel, D. J. (1991) An active modelling system for econometric analysis, Decision Support Systems 7 pp 315–328.

Dutta, S. and Bonissone, P. (1990) Integrating case based and rule based reasoning: the possibilistic connection, Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence, Boston.

Golding, A. and Rosenbloom, P. (1991) Improving rule-based systems through case-based reasoning, Proceedings of the National Conference on Artificial Intelligence, Los Angeles.

Hines, W. J. and Pinnes, E. L. (1988) Trends in the automation of service provisioning, IEEE Journal on Selected Areas in Communications, 6, pp 656–661.
Keen, P. G. W. and Scott Morton, M. S. (1978) Decision support systems: an organisational perspective, Addison Wesley Publishing Company, Reading, Massachusetts,
Nii, H. P. (1986) Blackboard systems (part one): the blackboard model of problem solving and the evolution of blackboard architectures, The AI Magazine, Summer 1986, pp 38–53.
Raghavan, S. A. (1991) JANUS, a paradigm for active decision support, Decision Support Systems 7 pp 379-395.
Sprague, Jr. R. H. (1986) A framework for the development of decision support systems, in Decision support systems: putting theory into practice (R. H. Sprague, Jr. and H. J. Watson, eds.) Prentice-Hall Inc., Englewood Cliffs, New Jersey.