A blackboard-based decision support system for the configuration of telecommunications services
S. Parsons, T. Brown, T. Chau & E.H. Mamdani
Department of Electronic Engineering, Queen Mary and Westfield College, Mile End Road, London E1 4NS, UK

ABSTRACT

The predicted expansion of the telecommunications market in Europe will place a premium upon the ability of providers of telecommunications services to efficiently configure systems that satisfy their customers' needs. The complexity of the domain will mean that configuration by hand will not be an efficient solution, and some kind of computer-based support will be required. In this paper we describe a decision support system that provides a possible solution to the problem of configuring telecommunications services.

1.0 INTRODUCTION

It is predicted [3] that the telecommunications market in Europe will undergo great changes in the next few years. There will be many more services from which customers may choose, and these services will be offered by a number of different organisations making the market highly competitive and giving the customer a choice not only of service, but also of provider. It is suggested that customers will make their selection of telecommunication services by looking for providers who can offer short delivery periods, accurate forecasts of delivery dates, competitive pricing and high quality of service. To satisfy these requirements, service providers will need to be able to allocate resources to customers efficiently, address specific customer needs, have accurate forecasts of delivery and optimise the use of their resources. Carrying out such a process is known as provisioning [7]. The diversity and scale of the networks and services in which such provisioning must be performed makes it impractical for humans to complete the task alone, whilst the semi-structured nature of the provisioning process means that it may not be completely automated. The solution seems to be the use of powerful decision support applications which can help a service provider's personnel in provisioning advanced communication services and in the deployment of resources to meet customer requirements.

The aim of our work is to demonstrate the applicability of decision support systems (DSSs) to service provisioning in an environment such as that described above. We are developing three decision support systems and using
the experience gained to develop a toolkit that will be used to produce further DSSs. This paper describes one of the systems but first we put this work in the context of the project and of service provisioning as a whole.

2.0 DECISION SUPPORT SYSTEMS IN SERVICE PROVISIONING

The process of provisioning a telecommunications service involves the analysis and presentation of large amounts of information. This includes surveying a large portfolio of services in order to identify the service which best matches a particular customer's requirement, as well as assigning and activating a variety of network components in order to deliver the service. The role of a DSS in this process is to interpret the information, taking raw data and turning it into something meaningful for a user to work on. Such a DSS is a supportive co-worker, who analyses and processes information presenting the user with suggested options, solutions and strategies and is thus an active partner in the problem solving process. This is essential in a domain as complex as that of telecommunication provisioning where the user needs as much assistance as possible.

It seems that successful DSSs will make use of a number of different technologies each of which will provide a part of the necessary problem solving functionality, information modelling and storage, human-computer interaction and problem solving control. Of particular interest to us is the combination of knowledge-based systems (KBS) and operations research (OR) techniques for problem solving. The field of operations research presents a vast array of algorithms and techniques that provide optimal solutions to well-defined problems. In contrast KBS techniques can produce acceptable solutions to ill-defined or incomplete problems and, typically, require less processing time than their OR counterparts. It is envisaged that the synergy of these technologies will provide powerful support systems that use a judicious mixture of optimal and acceptable methods.

In the DESSERT project we are developing three decision support systems which address different stages of the provisioning process. These are:

- the Customer Requirements Capture application which helps to establish what services a customer requires, and then produces a network independent technical specification of the required service,

- the Generation and Selection of Alternative Configurations application which takes a technical specification of a service in terms of its communication, processing and usage requirements and assigns the most appropriate resources to meet these requirements, and

- the Resource Scheduler application which allocates times and resources for jobs which have to be performed in order to start the services that have been requested.

It is the Generation and Selection of Alternative Configurations (GSAC) system that is the subject of the rest of this paper. Firstly we discuss in some detail the context in which GSAC will operate. Then we describe the way that the
system is employed in generating and selecting configurations before delving into the implementation detail of the system.

3.0 THE APPLICATION DOMAIN

In the telecommunications market, there are a number of possible relationships between those involved in the provision of advanced communications services (see Figure 1). The figure shows that, for instance, one service provider could buy services from a network provider, sell these on to another service provider who eventually sells a service to a customer. Each service provider adds value to a service by enhancing it in some way.

![Figure 1: Possible relationships between network and service providers](image)

Given the wide variety of provider relationships in the future telecommunications market, we have adopted a particular scenario within which to develop the GSAC application. This is illustrated in Figure 2. In developing this scenario we have assumed that a customer's requirements for a telecommunications service will place requirements upon the Service Access Box (SAB); the Access Network (AN); Specialised Service Provider Resources (SSPR); the Access Network/Transit Network (AN/TN) interface; and the
Transit Network itself. Each component needs to be either installed or configured in order to support a request for a service.

The main actors in this scenario are the customer, the network provider, and the service provider, and their areas of responsibility are delineated by the shaded areas in Figure 2. In this scenario, the service provider will use GSAC to determine how best to fulfil the customer’s service requirements, optimising the use of his own resources as well as his use of the network provider services. At the same time the service provider has to ensure the accurate and timely provision of the service to the customer.

The customer buys services from the service provider and these are delivered through the service access box. This terminates the connection to the customer’s premises (the access network) and supports interfaces to the customer’s premises equipment. The network provider in this scenario is a large public network operator who offers transit network communication services, that is long distance trunk connections over three different types of network: Metropolitan Area Network (MAN), Integrated Services Digital Network (ISDN) and Public Packet-Switched Data Network (PPSDN). The service provider owns, controls, and maintains resources which give a customer access to a service. Some of these may be either in the service access box, or attached to the access network or transit network. These resources are not just basic communication resources but also such features as encryption, format translation, and database backup.

4.0 AN OVERVIEW OF THE APPLICATION
Having described the scenario in which the GSAC application operates, we now introduce the application and the blackboard architecture around which it is implemented.

4.1 The function of the GSAC application
The GSAC application ensures that a customer's requirements can be met before any contractual commitments are made. In order to do so, a request for a service needs to be examined in depth in order to determine which telecommunications resources need to be assigned to that customer, and in what combination they must be configured. As input the application takes a customer's requirements in terms that are oriented for the service provider, but which are independent of any implementation technology. Such a request is evaluated taking into account the current configuration of equipment, the size of a customer's telecommunications traffic, and the usage pattern of the service provider's resources such as the service access box and the access network. In performing this evaluation, the GSAC application generates alternative configurations which fulfil both customer and service provider requirements, and assists the user in choosing the configuration that best matches the customer's needs.

The creation of the configurations is a complex task, especially because of the following considerations. Firstly, the technology independent input to the application needs to be mapped to sets of service and network resources. This process requires judgements to be made of the "best" solution and approach, and the translation of subjective measures such as quality of service into objective measures such as performance requirements. Secondly, the service provider will need the application to use a number of different optimisation criteria during the process of generating configurations as well as in the selection of the most appropriate configuration. These criteria include minimising the cost to the service provider, ensuring the greatest reliability of service and the highest security of data, and attaining the best fit with a customer's plans for future evolution. Other sources of complexity are the need to consider service access boxes, access networks, and access network/transit network interfaces for many different customer sites (all of which will place constraints on each other) and the need to optimise service provider resources, such as digital cross connects or multiplexers, across a number of different customers. Furthermore, there is no straightforward process for creating a configuration. The configuration of any part of a service will have an effect on, and be affected by, all the other parts so there is a considerable amount of work involved in ensuring the consistency of solutions.

4.2 The blackboard architecture
The basic architecture of the GSAC decision support system is as depicted in Figure 3. The central structure, which ties all the various components together, is a blackboard [8] which may be thought of as a global database in which the current, partial, and final solutions are recorded. Connected to the blackboard are the various problem solving modules which together provide the decision support functionality that is required to complete the configuration task. These modules are based on a mixture of technologies including operations research, rule-based reasoning, case-based reasoning [10], and constraint satisfaction. The modules access an Information base and are controlled by some problem solving control module.
The reason for organising the system around a blackboard is that it provides a clean method of integrating various AI and OR techniques into a single system. There are other systems that combine different problem solving methods such as case-based and rule-based reasoning [5] [6]. However, these often work by "shoehorning" one problem solving mechanism into the format of another, for instance by writing cases as rules, a process that can distort the data and destroy the advantages of the representation that is being shoehorned. The DESSERT blackboard approach does not require this to be done. Instead each module uses its own knowledge representation, for instance rules for the rule-based module and cases for the case-based module, and integration is achieved by having each module read and write to the blackboard using a common format.

![Diagram of the blackboard architecture of the GSAC decision support system](image)

**Figure 3. The blackboard architecture of the GSAC decision support system**

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 (KS) in blackboard parlance, monitors the blackboard for partial solutions to which it can contribute, creating a knowledge source activation record (KSAR) indicating its willingness to work on a particular solution. The control module co-ordinates the KSARs and selects a module based on knowledge about the module and the nature of the solution on which it wants to work. The module acts on the partial solution and updates the blackboard accordingly, then the cycle begins again. In deciding upon problem solving strategies the control module will use knowledge about the KSs to choose between them and to change their order of execution to obtain the best possible interaction. Typical criteria for making this choice are the estimated speed with which the KSs can perform their task, the quality of the solution and the task which the KS can perform. The criteria will be drawn from fundamental information about the tools and data gleaned from past experience. For instance given an operations research tool and a rule-based tool which can both work on the same partial solution it is clear that the OR tool is likely to give a better solution while experience might indicate
that at the particular stage of a configuration the rule based tool gives adequate results in a shorter time.

Since it is a decision support system, it is imperative that the GSAC system is as interactive as possible. Therefore, a major requirement is that the user should be able to contribute partial solutions to a problem and should be able to control problem solving. This goes against the traditional blackboard approach in that blackboards are usually allowed to run under their own guidance with no interference from a human operator. A user layer has been introduced to the blackboard to provide user interaction. The first function of the user layer is to provide a window onto the blackboard. Through this window the user can read information from every layer of the blackboard and can also write to any layer on the blackboard. Thus the user is seamlessly integrated into the system as another knowledge source, enabling her to do anything that a knowledge source can; namely, read the blackboard and modify the partial solutions that are on it. In addition, the user is given privileges not afforded other knowledge sources, being exempt from having to send a KSAR to the control module before being allowed to modify a partial solution on the blackboard, so that she may jump in ahead of the other knowledge sources. In addition to contributing to partial solutions, the user may help in determining the problem solving strategy through the user layer. She may question the system as to why a particular KSAR was selected for execution and is allowed to alter the agenda which lists the order in which the knowledge sources are executed. The user is thus permitted to interact with the control module and specify which module should be executed next.

As a result of the above mechanisms, the user and the decision support system feed off one another, achieving a kind of synergy in which their joint efforts produce greater results than either could produce alone [1]. For instance, the user would not have the ability to remember the large number of cases that the case-based reasoning module can draw upon, or be able to apply the weighted averaging of criteria that the operations research modules can. The computer could not provide the flashes of insight of the human operator, or the ability to relate the need to reduce cost to the selection of a different type of access network.

5.0 USER INTERACTION

As mentioned above, the fact that the GSAC application is a decision support system means that its operation is largely user-driven. That is the user initiates operations, deciding at a high level what part of the configuration will be tackled at a given time, and directing the efforts of the system to that end. Once this decision is made, the system takes charge using the opportunistic control of the blackboard mechanism to guide its efforts, though the user may contribute to the problem solving process and can interrupt the system or cause a different knowledge source to be employed. Once a particular part of the configuration has been completed, control is explicitly passed back to the user who chooses what to do next. In this section we look at the kinds of interaction that take place in more detail.
The initial task performed by the system is to display the services that have been requested, and which must be configured (Figure 4). Each service request is represented as a straight line between two points. A circle represents a termination at a customer site whilst a square represents a termination at the transit network. Thus non-switched services such as leased lines will be shown as straight lines between circles whilst switched services such as telephone or fax links will be shown as straight lines that run from a circle at the customer's site to a square representing the transit network connection. Using a mouse to click on the line representing a service, causes it to be expanded to show the detailed requirements obtained from the user in the technology independent form in which they are passed to the GSAC application.

**Figure 4. Displaying new service requests**

Clicking on the “Generate possibilities” button generates a window that displays the Feasible Topologies (see Figure 5, below), that is the set of possible network topologies that will satisfy the service requests. A particular service between two sites will typically require a transit network connection between the sites, and it is possible that each site may be connected to the transit network at more than one point. The set of feasible topologies represents all the possible options for connecting all the sites between which services are required. Where options exist they are represented by light coloured lines. Where no options exist dark lines are used. No optimisation of the service is performed at this level; it only the connections that might be used to implement the service. Furthermore, the connection is only described at the highest level, having as components the access network, transit network and service provider
resources, so that no consideration is given to how each of these components is implemented.

By clicking on the relevant part of the window the user can cause the "Access link", "Transit link" or "Service provider resource" configuration tasks to be executed. When initiated, each of these tasks will proceed to determine all the valid mappings from technology independent service descriptions into legal sets of components and resources as described in Section 6. Thus configuring an access link might involve a number of knowledge sources and the user working together using case-based and rule-based reasoning to select a set of components that implement an ISDN connection and a PPSDN connection. The user can then choose which of these is best using the multiple-criteria decision making tool [2], perhaps as a result abandoning one of the possibilities, before proceeding to carry out another of the configuration sub-tasks.

The user has control over the order in which the three sub-tasks are carried out, but it should not be assumed that, because they may be performed in any order, the three tasks are independent. The result of each sub-configuration can have an effect on the others. Choosing a particular access network configuration will constrain the possible transit link and service provider resource configurations with the system propagating constraints from one configuration task to the others. The cumulative effect of these constraints may be, for instance, to make a given transit network connection incompatible with a particular access network configuration. In this situation the user can choose to reconfigure either of the incompatible parts. As a result the process of determining the best feasible configuration is an iterative one, with the system assisting the user by automating the expansion of particular sub-
configurations enabling her to explore the space of all possible configurations faster than on her own.

6.0 KNOWLEDGE SOURCES AND THEIR INTERACTIONS

Having considered the operation of the GSAC application from the point of view of its user, we look at the system from the point of view of the components that carry out the configuration sub-tasks.

6.1 How a configuration is determined

One way of viewing the process of determining the configurations is as a series of mappings. The data arrives in the form of a number of sets of abstract service descriptions, known as abstract communication entities (ACEs), each of which corresponds to a service request. These are mapped onto the set of feasible topologies, known as service topologies (STs), a process which determines which resources will be required in each part of the network. The resources in each position are then aggregated and mapped onto a set of entities that describe the function of the components of MANs, ISDNs, PPSDNs, and other service provider resources such as encryption and data storage facilities. These are known as functional entities (FEs). Once the functional descriptions of given configurations have been tested against customer constraints (such as compatibility with certain pieces of terminal equipment) the functional description is mapped onto a physical description which identifies the actual components that will be used to provide the service. This physical description is a set of physical entities (PEs). The idea of including a functional description is to provide a stage at which the configuration is not dependent upon a particular network, minimising the amount of customisation needed to adapt the system to another service provider, or a new network implementation.

The mappings do not take place in series. Once the set of service requests is split into a set of feasible topologies, and it is decided which part of the configuration to work upon, a number of sets of competing tool components can run. For instance, one of the parts of access link configuration is Access Pipe Optimisation; the means of determining the number and type of the functional connection entities. The GSAC application includes both operations research and knowledge-based system components for doing this. The operations research tool uses linear programming to give an optimum solution, while the knowledge-based system gives a quick means of generating solutions that are often acceptable. Other software components will co-operate. For example, in order to evaluate a given configuration using the multi-criteria decision making tool, it is first necessary to run objective evaluator functions to determine the degree to which particular criteria are met, for instance, what the cost of the configuration is. Thus a user request for evaluation will trigger the multi-criteria decision making component which will call the necessary objective evaluation functions. However, in anticipation of the use of the decision making tool, objective evaluation functions may trigger themselves whenever they notice that a configuration has been completed.

There is also a certain amount of backtracking. As configurations are completed, they are checked against customer imposed constraints such as the requirement to use a certain piece of premises equipment. If configurations violate such constraints, then they are no good, and the computation must
backtrack to an earlier stage and perform the mapping down to functional and physical descriptions once again.

6.2 The GSAC knowledge sources
As discussed in Section 4, the GSAC application is implemented using a blackboard, and thus the modules discussed above are built as knowledge sources around that blackboard. Some of the knowledge sources are identified in Figure 6. Note that the blackboard is divided into layers. In the top layer the main data are ACEs, in the second they are service topologies, in the third functional entities and in the fourth physical entities (the user layer is not shown). This division is for efficiency, since knowledge sources that map between layers need only monitor changes on a single layer, and allows different control regimes to be implemented on different layers.

![Figure 6. The knowledge sources used in the GSAC application](image)

The knowledge sources are classified by function and are described below.

**Down-mapper Knowledge Sources:** These map data on one layer of the blackboard to data at another lower level.

ACE-ST: this knowledge source uses data about the ACEs that describe the connections between sites to create the basic connectivity pattern of the service topology.

ST-FE: these knowledge sources provide the mapping from a feasible topology to a set of functional descriptions. This knowledge source may generate more than one set of possible functional entities depending upon...
the characteristics of the connections in the service topology. There are three of these knowledge sources, one each for MAN, ISDN, and PPSDN.

**FE-PE:** these knowledge sources map from functional entities to physical entities, largely on the basis of what equipment is already present in the access network since the aim of provisioning is to allocate existing equipment rather than arrange for the installation of new equipment. Once again there are three of these.

**ACE-FE:** this uses case-based reasoning to provide a direct mapping from the list of abstract elements 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. Other modifications are carried out by the value-adder knowledge sources.

**Value-adder Knowledge Sources:** these knowledge sources operate on attribute values at a particular level of the blackboard.

**ND-LST:** this knowledge source fleshes out the basic service topology using information from ACEs such as those that represent encryption and help desk facilities.

**AST:** a knowledge source that takes the requirements heaped on different attributes of the service topology and aggregates them to come up with a complete set of requirements. This aggregation may be a simple addition of bandwidths for a datalink, or a more complex Erlang calculation to establish the necessary number of phone lines.

**MCDM:** a multi-criteria decision making knowledge source, based on fuzzy weighted averaging [2], helps the operator assess which of the sets of physical entities identified as possible ways of implementing the services that the customer requires, best serve 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 simple heuristics to establish the degree of security to complex cost estimation functions based on operations research methods.

**Backtracker Knowledge Source:** this knowledge source unwinds failed solutions, thus backtracking up the blackboard.

**MCPE:** if a configuration fails due to inconsistency with a customer's equipment, it may be necessary to reassess the functional entities by backtracking to this level and creating another set of FEs with the specific aim of reducing the requirements on the customer's premises equipment.

**Consistency-checker Knowledge Sources:** these knowledge sources check the consistency of solutions.

**CBR-CHK:** this checks the output of the case-based reasoner and patches the solutions where possible by making changes to the list of FEs using a set of heuristics developed from previous cases.
CPE-CHK: this operates at the PE level and examines sets of PEs corresponding to suggested solutions comparing the requirements placed on the customer's premises equipment and the capabilities of that equipment. If there is any discrepancy a flag is raised to trigger some corrective action.

**Up-mapper Knowledge Source:** this knowledge source 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 so that they may be used in the process of re-negotiating the services that the customer wishes to buy.

**Strategy-Suggestor Knowledge Sources:** these knowledge sources examine a particular partial solution and suggest strategies that might be adopted.

BT-PRIOR: This knowledge source prioritises the use of backtracking knowledge sources when there are many sets of physical entities which are inconsistent with the customer's premises equipment.

DEPTH: This knowledge source focuses the broad sweep of the conventional blackboard approach into search into a depth first goal-oriented search by prioritising those knowledge sources that work on a nominated partial solution.

**6.3 Evaluating the system**

Performing an evaluation of the GSAC application is rather difficult. The scenario for which the system was designed is not one that is realistic at this time; it was designed to be an approximation of the situation in five to ten years when the telecommunications market has become much more diverse. Thus it is not possible to take the system and try it out in its natural habitat (at least not for five years or so). However, it is possible to validate the system to ensure that it behaves correctly, and it is planned to carry out a validation later this year. For now we have to complete work upon a suitable methodology for ensuring the correctness of a system which uses a variety of problem solving techniques.

**7.0 SUMMARY**

This paper has described the Generation and Selection of Alternative Configurations application developed during our work on the DESSERT project. This is a system in which machine and user form a team, with the application providing active decision support [9] and [4] in the area of telecommunications service provisioning. 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 posed to it the system sets to work, using its battery of operations research and knowledge-based systems techniques to provide 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 decision-making tool, or she can get involved in the generation of a solution by changing one of the partial solutions as it sits
on the blackboard. When doing this the system stops to let her complete her interaction before continuing to amend the solution using its compiled expertise. This mode of working is thus completely flexible allowing the user’s skills to complement the ability of the machine and providing powerful support for decision making in a complex domain.

ACKNOWLEDGEMENTS

The work described in this paper was carried out during project R2021 DESSERT of the RACE program and funded by the Commission of European Communities. The authors are grateful to their colleagues from the other organisations in the project, Broadcom Eireann Research, BT, Dublin City University, Framentec-Cognitec, Inform GmbH, PTT Netherlands Research, SEMA Group, and Trinity College Dublin, for their help and advice.

REFERENCES