Constructing a virtual training laboratory using intelligent agents — **Source link** ↗

Timothy J. Norman, Nicholas R. Jennings

Institutions: Queen Mary University of London

Topics: Virtual training, Virtual Laboratory and Intelligent agent

Related papers:

- Multi-Agent Based Pedagogical Games

- Applications of intelligent agents

- A multi-agent architecture to support synchronous collaborative learning in an international environment

- An agent-based intelligent virtual learning environment for construction management

- Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)

Share this paper: 𝐟 🐦 in ✉

View more about this paper here: https://typeset.io/papers/constructing-a-virtual-training-laboratory-using-intelligent-1ajgiqi6x7

Constructing a Virtual Training Laboratory using Intelligent Agents

Timothy J. Norman and Nicholas R. Jennings
Department of Electronic Engineering, Queen Mary and Westfield College,
University of London, London E1 4NS, UK.
`{t.j.norman,n.r.jennings}@qmw.ac.uk`

This paper reports on the results and experiences of the Trilogy project; a collaborative project concerned with the development of a virtual research laboratory using intelligent agents. This laboratory is designed to support the training of research students in telecommunications traffic engineering.

Training research students involves a number of basic activities. They may seek guidance from, or exchange ideas with, more experienced colleagues. High quality academic papers, books and research reports provide a sound basis for developing and maintaining a good understanding of an area of research. Experimental tools enable new ideas to be evaluated, and hypotheses tested. These three components—collaboration, information and experimentation—are central to any research activity, and a good training environment for research should integrate them in a seamless fashion. To this end, we describe the design and implementation of an agent-based virtual laboratory.

Keywords: intelligent agents; virtual laboratory; telecommunications.

Biographical note: Tim Norman has been a post-doctoral research assistant at Queen Mary and Westfield College since November 1995. Before this time, he was studying for a PhD at the department of computer science, University College London on theoretical artificial intelligence.

Biographical note: Nick Jennings is professor of intelligent systems at Queen Mary and Westfield College. His research interests are in the theoretical and practical aspects of agent-based computing.

1 Introduction

One of the primary roles of universities and other teaching establishments is to provide good quality training environments in which students can learn about core ideas and can investigate new directions. This is especially true for postgraduate students. To this end, the Trilogy project [1] aims to improve the training of research students through the development of a virtual laboratory across three collaborating universities. The three institutions have a common research interest in telecommunications traffic engineering, although with different specialities and approaches. Thus if this broad expertise is pooled and made available to students, their research will benefit from the synergy that comes from accessing knowledge from, and collaborating with, members of other groups. Note although the Trilogy laboratory is tailored for students in the telecommunications domain, the underlying concepts and technology are more widely applicable and can easily be transferred to other domains of research.

Training postgraduate students involves a number of basic activities. Firstly, they may seek guidance from, or exchange ideas with, more experienced colleagues. Secondly, high quality academic papers, books and research reports provide a sound basis

for developing and maintaining a good understanding of an area of research. Thirdly, experimental tools enable new ideas to be evaluated, and hypotheses tested. Together these three components—collaboration, information and experimentation—are central to any research activity, and a good training environment should integrate them in a seamless and coherent fashion. Such integration is problematic at present because people and resources are widely distributed, and because researchers are not necessarily aware of the interests of others, or the information sources and tools that are available. In many cases, even if they are aware, utilising tools developed by others is prohibitively difficult (for a range of technical and non-technical reasons).

Against this background, one of the aims of the Trilogy project is to provide a comprehensive index of high-quality academic papers, reports, web sites and other sources of information that are of relevance to telecommunications traffic engineering. A resource of this kind benefits users in a number of ways. Research students are provided with a sound basis for investigation. Experienced researchers have a reference to core results in their field, and a forum for sharing new findings. However, the term "virtual laboratory" suggests more than just a repository for information; i.e. more than a digital library [2,3]. Effective research is certainly dependent on access to pertinent information, but in scientific and engineering disciplines, principles are illustrated and new hypotheses are tested through experimentation. Thus, the laboratory should also include access to experimental facilities. Experimentation benefits researchers in a number of ways. Research students are provided with access to tailor-made experiments to aid in learning about certain phenomena and experimental techniques. Experienced researchers may back up research results with simulations that illustrate their key ideas. In the latter case, good documentation and well-managed access to these tools is encouraged if there is a mechanism for others to use them. If a sharing and re-use culture can be established through the laboratory, the effort involved in developing tools or experiments will continue to benefit a research institution after the project for which they were developed is complete. In this sense, the laboratory can also act as a corporate knowledge repository.

Motivated by these issues, the Trilogy system uses intelligent agents to present information and to provide access to tools in response to a user's expressed requirements (see also the UMDL [4]). As discussed in the following section, agents may also proactively suggest other possibilities that a user may not have considered. In addition to acting for users, agents manage the resources in the system, and enable existing information and experimental resources to be more easily integrated (see Section 2). The Trilogy system itself is presented in Section 3. Section 4 discusses the design decisions involved in developing the virtual laboratory. Finally, Section 5 presents our conclusions and indicates future research directions.

2 The role of agents in the virtual laboratory

A number of issues must be addressed in the design of a system to support the learning of research students, and the continued research of more experienced users. First, there are issues relating to the presentation of information and resources to the user.

- The system should, whenever possible, tailor its responses to the interests and experience of each user. For example, a novice should be provided with introductory material relating to their topic of interest, whereas a more experienced researcher may wish to be presented with a list of recent publications in appropriate journals.

- The system should suggest relevant information to the user. For example, if the user plans to use a particular experimental tool, the system should proactively identify tutorials, user manuals, and similar experiments that have already been performed

using that tool. Such proactive behaviour is warranted because the user may be unaware that such information is available.

Second, there are issues relating to the management of information and experimental resources within the system.

- Access to scarce resources must be managed. For example, it may only be possible for a particular experimental tool (e.g. a network simulation tool) to be used for one experiment at a time. However, the demand for access to this tool cannot be predicted in advance. Therefore, it is essential for the system to manage access to the tool by scheduling experiments in an appropriate and flexible manner.

- It is important for the system to be easily extensible as new information and experimental resources become available. However, different search engines may have different query formats, and each experimental tool may have different initialisation procedures. Therefore, an overarching integration framework is needed to cope with the diverse requirements of the constituent components.

Given both sets of issues, we decided to adopt an agent-based solution. The rationale and justification for this choice is explored in the remainder of the section. Before we do this, however, it is essential to define what we mean by an agent and a multi-agent system. For us, an agent is an encapsulated entity that is capable of responding to changes in its environment, acting in anticipation of such changes, and interacting with other agents to solve dependencies between them [5]. A system that consists of multiple agents that interact and that may co-ordinate their action in solving individual or joint problems is referred to as a multi-agent system.

Now we turn to the justification. First, the virtual laboratory is an inherently distributed system. This means that a system of multiple, interacting, autonomous agents is a natural model of the way the resources and users are distributed among the collaborating institutions. In addition to this, there are specific advantages with respect to the virtual laboratory as opposed to more traditional distributed information systems. With reference to the presentation of information and resources to the user, agents are appropriate because:

- The autonomy of an agent enables it to encapsulate the interests of a particular user, and tailor its responses to reflect those interests.

- The fact that an agent can anticipate the user's needs means that it can act to identify pertinent information of which the user may be unaware.

- If multiple users of the system have similar interests, agents can exploit user profile information to make them aware of one another [6].

Agents are also appropriate because they have the ability to manage scarce resources, and because they allow existing systems to be integrated in a comparatively clean and straightforward manner.

- The ability of agents to plan their activities ahead makes them appropriate for the management of scarce resources. The agent managing such a resource will monitor its operation, maintain a schedule of activity, and respond to requests for the use of this resource with some indication of when it will become available.

- The interactions between agents acting on behalf of a user and those managing resources can be sophisticated. For example, an agent that is instructed to obtain access to a scarce resource and the agent managing that resource must take into account the requirements of the user (e.g. when an experiment is to be performed by) and the other tasks that are scheduled (e.g. other experiments for other users) in

arranging for access. This requirement appeals to the social nature of agents since it involves both coordination and negotiation.

- For existing systems to be made available to the users of the virtual laboratory, they must provide an interface that is consistent with other resources, and that supports the flexible interaction required. Such resources may be "wrapped up" as agents to provide this flexible and consistent interface [7, 8].

A final advantage in the use of agents is their ability to record, maintain and communicate information *about* the system, as well as the information within the system. In particular, to satisfy a user's request, the agents within the system must maintain a record of the information repositories and software tools available, and how appropriate they are for various tasks. This "mediation" of system information is useful for any large, distributed information system, and it is especially useful if the availability of resources may change over time. Wiederhold [9] proposed the concept of a mediator in information system architectures as a way of managing the volume of information within such systems. Put simply, a mediator is an agent that develops and maintains an abstraction of the information or resources within the system. For example, a mediator may maintain information about software simulations of ATM routing protocols in telecommunication networks. Another mediator may then maintain a record of what mediators provide information about ATM software simulations. In our case, mediators occupy a distinct, active layer between user-oriented information processing and resource management.

Having made the case for an agent-based approach to the design and implementation of the laboratory, the following section details the Trilogy system architecture.

3 The Trilogy system architecture

The Trilogy virtual laboratory is designed and implemented as a multi-agent system, in which agents play one of three distinct roles: a personal assistant agent (PAA), a mediator or a resource agent (RA) (figure 1).

Our agents interact through an agent communication language [5], that enables them to, for example, "`ask for`" information about experimental tools, "`tell`" another that agent x can simulate certain types of traffic in a telecommunications network, "`propose`" that an experiment is performed at 5pm, and "`counter-propose`" that it be done at 5.30pm. Each message passed between agents is expressed in this language. Therefore, each agent must be able to respond to such messages. The use of a high-level communication language is the key to incorporating existing systems; whatever interaction protocol they require, the RA will translate instructions and information to and from the application-specific language and the Trilogy agent communication language.

Users of the laboratory interact with their PAAs via an http server. The particular server used is the Common Lisp HyperText Transfer Protocol server [10], which provides sophisticated methods for adapting the presentation of information to the user. Messages in the agent communication language are transported via the CORBA-compliant distributed object platform, Orbix [11]. Each agent is thus an object that is registered on this platform, and that provides methods that correspond to the primitives of the communication language.

3.1 Personal Assistant Agents

Personal assistant agents (PAAs) represent the interests of users within the system; there is one PAA for each user. The user specifies their interests (or profile) by giving a weighting to certain key concepts in the domain of interest; in this particular case tele-
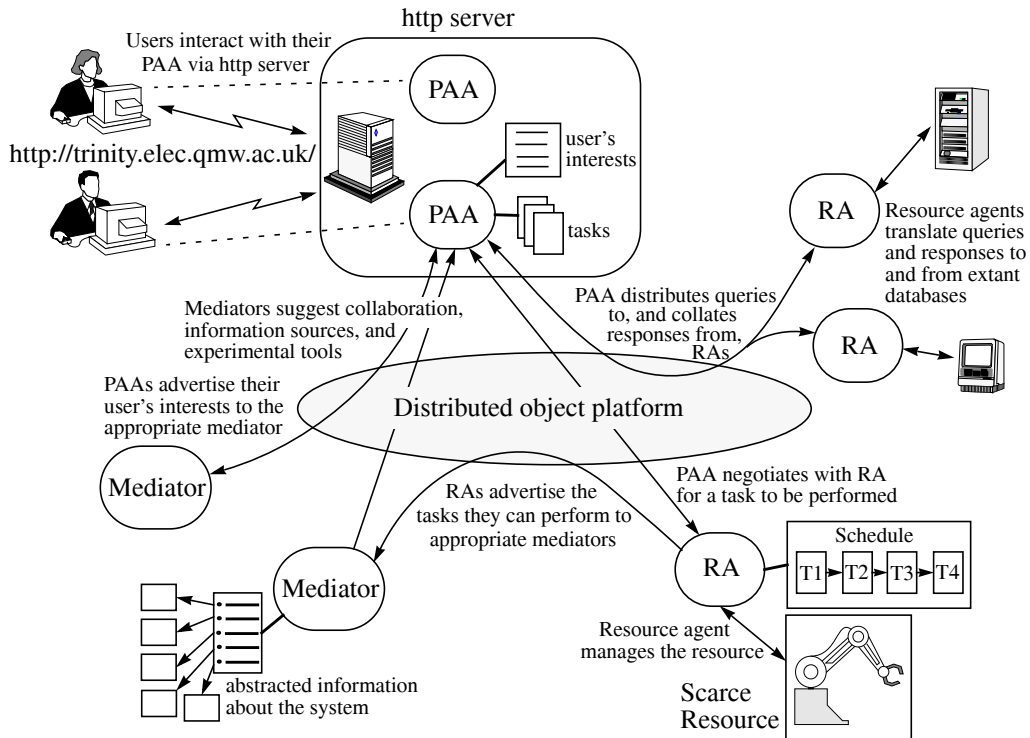
Figure 1. The Trilogy system architecture.

communications traffic engineering. These concepts are specified in an ontology [12] that defines the relationships between, and a characterisation of, each concept as a vector of keyword, weighting pairs. For instance, the concept "ATM Simulation" is related to a number of other concepts including "Burst level simulation" and "Cell level simulation", and is characterised by the vector {("simulation", 20) ("simulator", 20) ("experiment" 10) ("testbed" 10) ("tool" 5)...}. Together, the user profile and domain ontology provide the PAA with a basis for tailoring its responses to the user, and suggesting alternatives that may be of interest, but that have not been explicitly requested. For example, if the user requests information about burst level simulation experiments, the PAA will rank the options presented on the basis of the profile. Thus, the same query by different users may result in different results being presented.

In satisfying a request, such as obtaining information about burst level experiments, the PAA uses its knowledge about the system and about strategies for searching for information. For instance, the PAA may know of a particular resource agent that has been used in the past, which may provide a service that fits the user's requirements. This agent may then be contacted directly for information about the service. However, in other cases it may be more appropriate for the PAA to contact a mediator that maintains information about the experiments that are available in the laboratory. The mediator will then direct the PAA to either a number of resource agents, or a more specialised mediator.

In addition to questions about how the agent goes about achieving the user's goals, there are a number of user interface issues that need to be addressed. These are discussed in greater detail in Section 4, but with reference to figure 1, there is an important issue concerning the way in which a user interacts with their PAA. Generally speaking, the PAA should provide a consistent and, as far as possible, familiar interface to the resources that are available. It should also provide a natural extension to existing on-line information resources. For these reasons, a user interacts with the Trilogy system via a standard WWW client. Once a user has logged onto the Trilogy system, any

request for information or other instruction is forwarded by the WWW server to the appropriate PAA. The agent then provides results and other feedback to the agent by manipulating the presentation of information to the user by the web server.

3.2 Mediators

Mediators provide a number of intermediate information services to agents within the Trilogy system. For instance, a mediator exists for the purpose of maintaining information about simulation tools in the virtual laboratory. (Note there is an analogous mediator containing data about the laboratory's information resources). Each new tool that is added to the system is advertised to this mediator by its resource agent. The tool is indexed in the mediator by the concepts and keywords that characterise it. This information is then used by the mediator in responding to requests for information about which simulation tools are available. Similarly, if a tool goes off-line, the advert is withdrawn by its resource agent, and the mediator updates its representation of the system state.

The third Trilogy mediator records information about the interests of users. This information is derived from each user's profile, and is used in a number of ways. A close match between the interests of two users indicates that they may profit from collaboration (a match-making task [6]). Additionally, if there exists a user with a broader range of interests than another, then those wider interests may be suggested to the PAA of the other user (a collaborative filtering task [13]).

3.3 Resource Agents

Resource agents (RAs) provide a quick and effective mechanism for adding an experimental or information resource to the Trilogy system. Each such resource may provide a number of *services* to the system. For example, a simulation tool may provide a number of pre-set experiments, as well as access to an interface for configuring novel experiments.

For many of the resources in the Trilogy laboratory, access to the available services cannot be granted immediately. For example, the ATM testbed resource can only run a single experiment at any one time; similar considerations also apply to some of the software simulations. Thus, the resource may not be available when it is requested by a user's PAA. Suppose, that the user wishes to perform one of the experiments provided by this resource. It is now Tuesday, and the results are required before Friday. The PAA is charged with negotiating for the experiment to be performed within this time constraint; i.e. before Friday. The RA will also have time constraints. There are other experiments to be performed, each with specific time requirements such as deadlines, and each experiment may require different lengths of time for configuration, execution, etc. Through negotiation, the PAA and RA agree (if possible) on a time for the experiment to be performed so that the user's deadline is met, and no other constraints on the RA's activity are broken. Once agreed, the necessary configuration information is forwarded to the RA, and the service is initiated at the appropriate time.

A resource is an object registered on the CORBA-compliant platform [11]. It is managed through a simple interface. The resource must support interfaces to indicate the services it provides, the time requirements of each service, the inputs required and the outputs produced, and so on. It must also support an interface that initiates a service. This interface returns a reference to a service object that can accept the inputs required, return results and indicate its status (i.e. waiting for input, active, suspended, failed, etc.). Given that the resource and service interfaces follow their specification, generating a resource agent to manage this resource is a simple matter. A configuration file is written that specifies the name of the resource agent, the name of its resource,

and the names of any mediators to which its services are to be advertised. An example configuration file is given below.

```
FACTS:
    my_name_is "RA:ATMTestbed";
    my_resource "Resource:ATMTestbed";
    initial_acquaintance "themediator:experimentMediator";
```

On initiation, the RA queries the resource for information about the services it provides, and advertises these services to any agent of type mediator with which it is acquainted. The agent then waits for requests from other agents for it to provide its services. This agent manages all interaction with other agents through the agent communication language, maintains a schedule of its tasks, and executes services according to that schedule. The resource agent is implemented using a modified version of UM-PRS (a C++ implementation of the Procedural Reasoning System [14] developed at the University of Michigan [15]).

4 Rationale for the system design

Although the broad separation of concerns in the Trilogy system architecture is comparatively standard, there are two design decisions that are worthy of a more in-depth discussion. These are the balance of information processing effort between PAAs and mediators, and the balance between the autonomy of a PAA and the control of a user over its activities.

An important decision in designing a multi-agent system for the task considered in this paper is what information processing emphasis should be put on the mediator level. As indicated in Section 3.2, mediators are essential for tasks such as maintaining information about the resources available. Within the Trilogy system, there are mediators that provide information about experiments, information about literature search engines, and information about the interests of users. This meta-information aids a PAA in satisfying a user's requests. However, it is feasible for mediators to provide problem-solving abilities as well as meta-information. For example, the Ahoy! search engine [16] uses other Internet resources such as email services, general search engines and knowledge about the structure of specific web sites to locate WWW home pages for individuals. A similar agent could also be deployed in a virtual laboratory to provide a summary of the responses to a query from a number of existing search engines. For example, the task planning agent within the University of Michigan Digital Library (UMDL) [17] uses the content of a user's initial query to establish a team of agents that are appropriate for the satisfaction of that query. It then coordinates the activity of each member of this team.

Suppose then that such an agent is deployed in the virtual laboratory. This agent will use information about resources available from other mediators, and strategies (or plans) for achieving the user's goals. For example, depending on the goal of the user, introductory tutorials and lecture notes may be more appropriate than experimental techniques or theoretical models. The specialisation of information resources (e.g. using one that is specialised in maintaining a collection of lecture notes and past examination papers) can then be exploited to improve the quality of the solution. In this case, part of the PAA's problem solving role is delegated to this mediator. There are a number of advantages to this approach. First, the selection of a good plan can benefit from the experience of achieving similar goals. For example, if the resource that provides lecture notes and other introductory material was useful in the past for a particular goal, it is likely to be used again for a user with similar requirements. Second, the

mediator can exploit the similarities in the goals of different users to make the most efficient use of its time, and that of the system resources. However, considering the interactions (positive or negative) between different goals makes the selection of (and possibly the combination of) plans more difficult.

A distinct disadvantage of relying on shared problem solving resources in this way is that the mediator concerned may become a bottleneck in the system. As the number of goals that are to be achieved increases, and the number of potential interactions between these goals increase, the complexity of the planning problem significantly increases [18]. This problem can be alleviated to a certain extent through the replication of mediators. As the load on the mediator increases, it makes a copy of itself (or a clone [19]), that can provide the same service. As described by Shehory et al. [19], recognising when to clone is a non-trivial problem, as is deciding if and when a clone is no longer required. Furthermore, the most effective distribution of effort between these agents is also difficult. It is not always possible to determine whether two goals (positively or negatively) interact before plans are selected to achieve them. In fact, much work is required in modelling and understanding large-scale, self-organising multi-agent systems [20].

Within the Trilogy system, the selection of strategies to achieve a user's goal is performed by their PAA. In this way, the PAA can tailor its strategies more closely to the requirements of the user. For example, while the agent is searching for experimental tools to satisfy a user's goal, it may find a reference to a document that is relevant to another one of the user's interests. Because the PAA has access to the user's profile, it may recognise the relevance of this document. Furthermore, the PAA may establish collaborative relationships with other users with similar interests. Through such a relationship, information and experimental results can be shared (with the approval of the users concerned). Hence, it is possible through explicit collaboration between users to enable plans to achieve their individual goals to be combined.

The second issue discussed here is the balance between the autonomy of a user's PAA and the control that a user has over its agent's activities. A personal assistant agent is responsible for ensuring that the tasks it has been set by its user are completed, that feedback is provided to the user as to the status of each task, and that its responses are tailored to the interests of that user. In acting on behalf of a user, the PAA needs to make decisions about: how to identify agents that can contribute to the completion of a task, how to arrange for appropriate services to be provided, how to coordinate these activities and the information flows between them, and how to combine the results for presentation to the user.

By means of an illustration, suppose a researcher wishes to test a hypothesis about the behaviour of a specific network configuration. The user's PAA is presented with this as a problem to solve. The PAA will know the name of the mediator that has information about which software tools are available and so it will query this mediator about the appropriate tools. The PAA may also use other resources, such as an information repository to search for information regarding the use of each candidate tool. In such cases, the PAA will compile the results and provide a number of options to the user. Thus, it can be seen that the user's personal assistant agent not only satisfies the explicit request, but that it may also proactively search for information (e.g. user manuals and similar experiments that have already been performed) regarding each option so that the user can make a more informed choice. In figure 2, a number of candidate services are displayed to the user. The user can then choose to read information about these services before selecting the most appropriate. Once selected, the PAA will go ahead and arrange for the service to be provided, giving feedback to the user whenever appropriate. In this case, feedback is provided by indicating the status of the service, and by providing a summary when requested.
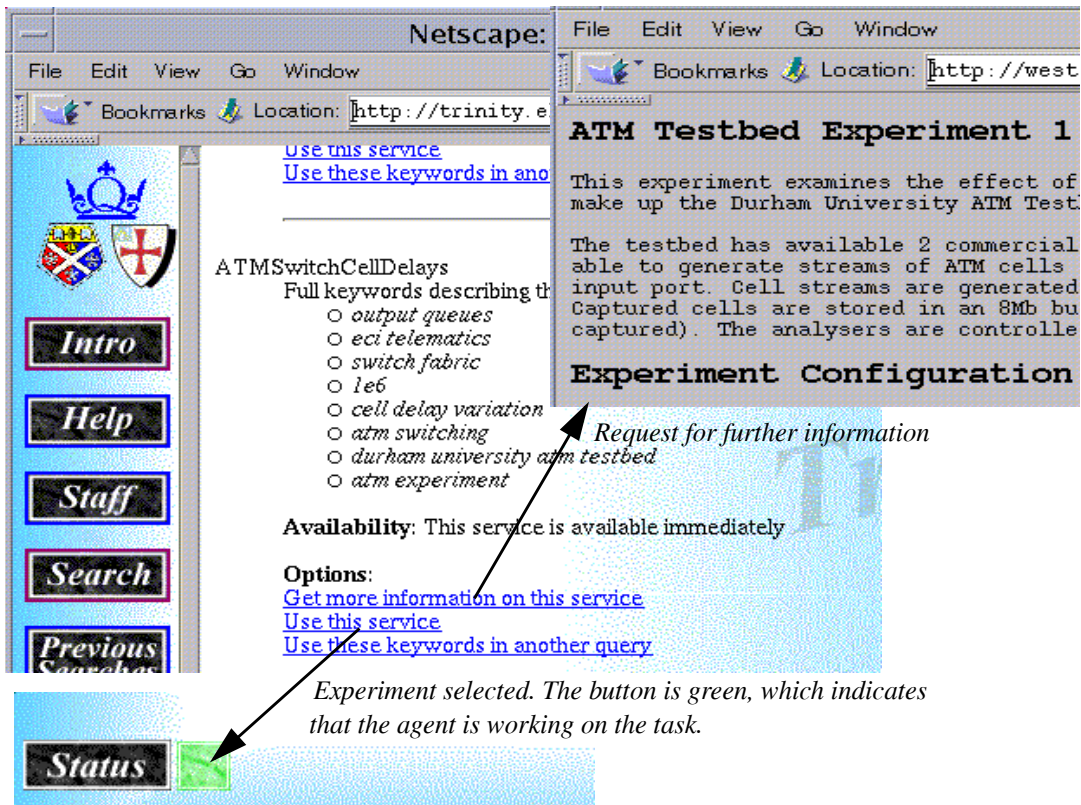
Figure 2. Presentation of possible experiments to the user.

Continuing with the scenario, imagine the chosen experiment requires some configuration by the user. For instance, the experiment is to be run on an ATM switch network (a hardware simulation tool situated in Durham University). In this experiment, two traffic sources are to be sent through a single switch, and depending on the characteristics of each traffic source, the behaviour of the switch will change. In figure 3, the status of the task changes to red, indicating that the PAA requires some decision or information from the user. When the user responds to this request for attention (by clicking on the red status button), the configuration of each traffic source (or stream) is requested. FWherever possible, the PAA seeks to automatically provide information on

Figure 3. Information required from the user for a specific experiment configuration.

behalf of its user and to offer default values for certain options. For instance, the user's name is an input that is often required for a service and this is provided automatically by the PAA. In our specific example, a default filename for the results is also specified by the agent; although the user is naturally free to change this.

This example illustrates the importance of striking a good balance between an agent acting autonomously, and giving the user a feeling of control over their interaction with the laboratory. Specifically, a user is given a feeling of control by: (i) presenting related information to enable the user to make informed choices; (ii) providing feedback as and when required; (iii) referring back to the user when information or confirmation of a decision is required; and (iv) allowing the user to retain the right of veto over an agent's activities.

5 Conclusion and future work

This paper has described the rationale, design and implementation of the Trilogy virtual laboratory. The laboratory uses agent technology to present a personalised and coherent training environment for the domain of telecommunications traffic engineering. This environment enables students to discover relevant information, to gain access to a wide range of experimental tools, and to share their insights and experiences with other members of the Trilogy user community. At this time, the laboratory is being tested by new research students at each of the collaborating institutions.

The development of the laboratory has uncovered a range of issues that require further investigation. Chief among these are the need to ensure our system architecture is scaleable and the need to facilitate richer collaboration between the students involved in the laboratory. In the former case, we are looking at systems that can dynamically self-organise to better fit with changing scaleability requirements. In the latter case, we are looking at the conceptual basis of cooperation between mixed groups of humans and artificial agents.

References and notes

[1] The Trilogy project is a collaboration between The School of Engineering, University of Durham, The Department of Electronic Engineering, Queen Mary and Westfield College, and The Department of Electrical and Electronic Engineering, University of Strathclyde. The project is funded by EPSRC under grant number GR/L09714.

[2] Nürnberg, P. J., Furuta, R., Leggett, J. J., Marshall, C. C. and Shipman III, F. M. (1995). Digital Libraries: Issues and Architectures. In <u>Proceedings of the Second Annual Conference on the Theory and Practice of Digital Libraries</u>.

[3] O'Leary, D. E., Kuokka, D. and Plant, R. (1997). Artificial intelligence and virtual organisations. <u>Communications of the ACM</u>, **40**(1), 52-59.

[4] Durfee, E. H., Kiskis, D. L. and Birmingham, W. P. (1997). The agent architecture of the University of Michigan Digital Library. <u>IEE Proc on Software Engineering</u>, **144**(1), 61-71.

[5] Jennings, N. R. and Wooldridge, M. J. (1995). Intelligent Agents: Theory and Practice. <u>The Knowledge Engineering Review</u>, **10**(2), 115-152.

[6] Foner, L. and Crabtree, I. B. (1996). Multi-agent matchmaking. <u>BT Technology Journal</u>, **14**(4), 115-123.

[7] Jennings, N. R., Varga, L. Z., Aarnts, R. P., Fuchs, J. and Skarek, P. (1993). Transforming Stand-alone Expert Systems into a Community of Cooperating Agents. <u>International Journal of Engineering Applications of AI</u>, **6**(4), 317-331.

[8] Genesereth, M. R. and Ketchpel, S. P. (1994). Software agents. <u>Communications of the ACM</u>, **37**(7), 48-53.

[9] Wiederhold, G. (1992). Mediators in the architecture of future information systems. <u>IEEE Computer</u>, **25**(3), 38-49.

[10] Mallery, J. C. (1994). A Common LISP Hypermedia Server. In <u>Proceedings of The First International Conference on The World-Wide Web</u>.

[11] Baker, S. (1997). <u>CORBA distributed objects using Orbix</u> ACM Press/Addison-Wesley.

[12] Gruber, T. R. (1993). A translation approach to portable ontologies. <u>Knowledge Acquisition</u>, **5**(2), 199-220.

[13] Maes, P., (1994) "Agents that reduce work and information overload" <u>Communications of the ACM</u>, **37**(7), 30-40.

[14] Georgeff, M. and Lansky, A. (1987). Reactive reasoning and planning. In <u>Proceedings of the Sixth National Conference on Artificial Intelligence</u>, 677--681.

[15] Lee, J. Huber, M. J., Durfee, E. H. and Kenny, P. G. (1994). UM-PRS: An implementation of the procedural reasoning system for multi-robot applications. In <u>Proceedings of the AIAA/NASA Conference on Intelligent Robotics in Field, Factory Service and Space</u>, 842-859.

[16] Shakes, J., Langheinrich, M. and Etzioni, O. (1997). Dynamic reference shifting: A case study in the homepage domain. In <u>Proceedings of the Sixth International World Wide Web Conference</u>, 189-200.

[17] Vidal, J. M. and Durfee, E. H. (1995). Task Planning Agents in the UMDL. In <u>Proceedings of the CIKM-95 Workshop on Intelligent Information Agents</u>.

[18] Bylander, T. (1994). The computational complexity of propositional STRIPS planning. <u>Artificial Intelligence</u>, **69**(1-2), 165-204.

[19] Shehory, O., Sycara, K., Chalasani, P. and Jha, S. (1998). Increasing Resource Utilization and Task Performance by Agent Cloning. In A. Rao, M. Singh and M. Wooldridge (eds.), <u>Intelligent Agents V</u>, Springer-Verlag.

[20] Turner, P. J. and Jennings, N. R. (1997). On Scaleability of Information Management Agents. In <u>Proceedings of the European IT Conference</u>.