

Using Case-Based Reasoning to Overcome High Computing Cost Interactive Simulations

Javier Vázquez-Salceda, Miquel Sànchez-Marrè and Ulises Cortés

Knowledge Engineering & Machine Learning Group, Technical University of Catalonia, Campus Nord-Edifici C5, Jordi Girona 1-3,
08034 Barcelona, Catalonia, EU
{jvazquez, miquel, ia}@lsi.upc.es

Abstract. This paper describes an innovative usage of Case-Based Reasoning to reduce the high computing cost derived from running large interactive simulation scenarios within the framework of advanced training systems. The paper discusses the ideas, implementation and a preliminary evaluation within the framework of the European IST research project A-TEAM.

1 Introduction

Over the last decade an important progress has been made in the Case-based Reasoning (CBR) field. Specially, because problems are more clearly identified, and research results have led to real applications where CBR performs well.

Case-Based Reasoning paradigm provides advanced intelligent training systems with a great learning flexibility. In advanced environmental training systems, the use of simulation exercises is a common feature. Providing the trainees with realistic skills in emergency management is an important learning feature of the advanced environmental training systems. Simulation exercises are good tools to practice environmental emergency management in a real scenario without causing damage to the environment, and complement other static learning issues of the pedagogical components of a training system. Commonly, simulation exercises have a major disadvantage: the computer time consumption is very high, as the simulation exercises usually involve complex mathematical models that are implemented with very time-expensive computer algorithms, and complex picture images, generated by highly time-consuming computer algorithms.

At this point, the role of the Case-Based Reasoning and its derived technologies appears as a possible solution. The CBR module can retrieve the appropriate previously computed simulation exercise/s according to the pedagogical issues of the learning units of an advanced intelligent training system, and/or according to the

degree of trainees' expertise. The CBR module can pre-compute and store in a Case Library the most representative simulation exercises in off-line computation. Afterwards, in an interactive on-line learning session with a trainee, the CBR module can retrieve the most appropriate set of simulation exercises avoiding huge computation time delays. Not only the CBR module can retrieve appropriate cases, but also it can adapt some retrieved cases to fit a new required simulation exercise foreseeing the simulation results, by means of some domain knowledge. Therefore advanced intelligent training systems will become a more interactive, pedagogically flexible and user-friendly training systems.

In Environmental sciences, CBR has been applied in different areas with different goals, because of its general applicability. It has been used in information retrieval from large historical meteorological databases (Jones and Roydhouse, 1995), in optimisation of sequence operations for the design of wastewater treatment systems (Krovvidy and Wee, 1993), in supervisory systems for diagnosing and controlling WWTP systems (Sánchez-Marrè et al, 1997a), in decision support systems for planning forest fire fighting (Avesani et al, 1995), in case-based prediction for rangeland pest management advisories by (Branting et al, 1997), or in case-based design for process engineering (Surma and Brauschweig, 1996).

In the environmental emergency management there have been some works, especially within the CBR planning field, such as the works of (Avesani et al., 2000; Ricci et al., 1999). But the merging of interactive simulation and CBR have not been exploited too much. There is a project (Sadek 2001; Sadek et al., 1999) about the development of a Prototype Case-Based Reasoning for Real-time Freeway Traffic Routing. This project developed a prototype Case-Based Reasoning Decision Support System for the on-line management of freeway traffic in the presence of incidents. The system uses input data from loops and closed-circuit TV cameras to develop optimal routing strategies that are to be relayed to travellers via variable message signs (VMS) and highway advisory radio (HAR). Also there some work in the training and education within the medical domain using CBR techniques such as the work of (Bichindaritz and Sullivan, 2002).

The paper is organised in the following way. Section 2 describes the framework of the A-TEAM project. In Section 3, the case structure within the system is detailed. Section 4 presents the case library interactions. Section 5 is devoted to the design and development of the CBR server. An application of the system in a particular domain is detailed in section 6. Finally, in Section 7 conclusions and future research directions are outlined.

2 The A-TEAM project

The CBR Module is one of the components of the A-TEAM System that provides information to the RTXPS module, as depicted in fig.1:

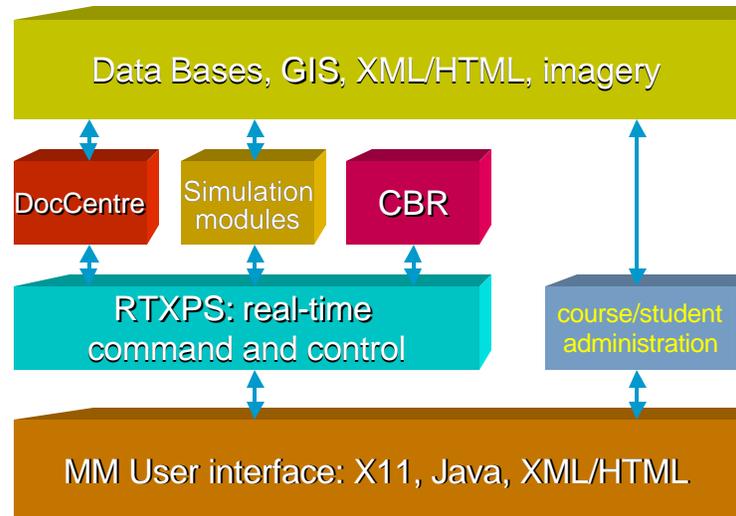


Fig.1. Overview of the A-TEAM modular architecture

The A-TEAM system provides to the trainee a multimedia course structured in lessons with text, images and short videos, where the evolution of the student is tracked by means of some tests. Most of contents of the courses (the lesson's text, images, videos and the multiple-choice tests) are provided to the RTXPS module by DocCentre. But there are other kind of tests that make A-TEAM system innovative: the interactive emergency situations.

Interactive simulated emergency situations are a special kind of tests where the trainee, alone or with other trainees in a shared scenario, has to solve an environmental emergency that is shown in their display. The trainee has to take the proper decisions to manage the emergency. These kinds of tests evaluate the trainee skills in emergency management, and are one of the key features of the A-TEAM system. Some of these emergency situations are completely set by the content provider, that is, the one who builds the course material. But there are times that the trainee is allowed to change some of the parameters involved in the emergency (such as wind speed, wind direction, wind temperature and so on) in order to see the influence each change in one of them has in the emergency.

Some of the emergency situations can be computed on-line by the Simulation Modules and delivered directly to the RTXPS system, but some others are so complex that it takes several minutes to compute them, so they shouldn't be generated on-line. In

order to keep the student focused in the lesson, it has been decided to avoid delays in computation of complex scenarios by means of having them pre-computed and stored in a way that lets a fast recovery of the information when RTXPS makes a request of one of those pre-computed situations.

One option to store the information of situations could be a Relational Database. It is a widely used technology to store and recover information in several information systems. However, the recovery is made by exact match of the values of the query. So using a Database in A-TEAM could lead to 2 alternatives:

- Pre-compute any possible scenario that a student could need for his training. This first option is unfeasible when there are emergency scenarios partially driven by the student's choice in the parameters, as the set of possible situations to store and retrieve would grow to million or even more combinations of those parameter values.
- To let only the content provider or the student to choose emergencies that currently are stored in the database. That is, that the queries made by the RTXPS to the CBR Module should match the exact values of the situations stored. This option makes that if there's a query that does not matches exactly with any pre-computed simulation, then no result is given.

As we saw in section 1, one of the main capabilities of a CBR system is the recovery of the best results, the ones that better fit with the values provided in the query. Instead of the "exact match" criteria, CBR systems use a "*similarity*" criterion (Núñez et al., 2003) that can be defined for each domain. Similarities allow to recover not only the exact result for the query (if exists) but also recover the most similar ones. In our CBR system *L'Example* similarity measure has been used (Sánchez-Marrè et al., 1998).

So the role of the CBR system is as storage facility for the pre-computed simulations so complex that they cannot be made on-line. In this new approach the pre-computed simulations are stored as cases in the Case Library. The queries that came from the RTXPS system are attended by the CBR module, which seeks for the most similar, pre-computed simulations to the one the RTXPS requests.

2.1 Integration of the CBR Module with the other A-TEAM modules

The CBR Module is integrated in the A-TEAM system as an independent module interacting with other modules in the system. These interactions of the CBR Module with the other modules can be divided in two steps (see figure 2):

- off-line: the Simulation Modules can provide the CBR with all the pre-computed data about complex emergencies (accidental spills, transportation accidents, fire, explosion, atmospheric dispersion) that could be useful to be recovered on-line.
- on-line: for each request from the RTXPS module, the CBR gives the best simulation according to a previously established similarity criteria.

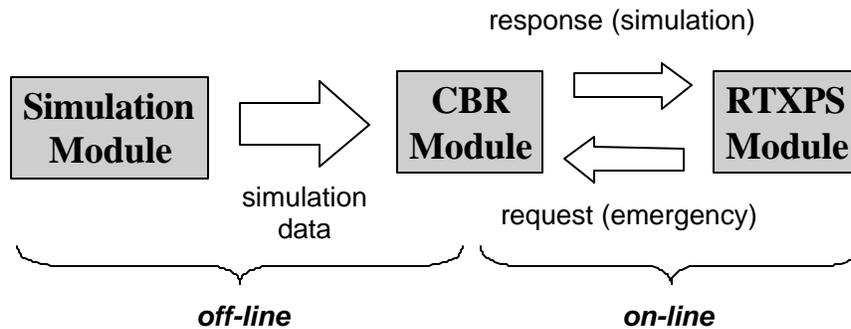


Fig. 2. Interactions of the CBR Module with the RTXPS module and the off-line Simulation Modules

Off-line interactions among the simulation modules and the CBR module are made by e-mail by means of a predefined format. On the other hand, the way on-line interactions are done is based on the work done in the WP06 (Integration and Implementation), that defines the A-TEAM system as a network-distributed service composed by:

- The Main Server, including the RTXPS system, the DocCentre System and the Data-bases
- The on-line UDM Simulation Server
- The CBR Server

The communication among all the distributed parts is made through the World Wide Web by means of an HTTP protocol. This distribution of computation allows that hard computational issues such as the emergency simulation or the CBR can be placed in powerful machines, speeding-up the response and decreasing the hardware needs for the end-users' platforms.

In the whole architecture (shown in figure 1), the CBR Server is the part of the A-TEAM system that does the CBR functionalities explained above. It is hosted in a powerful machine to a) store all the pre-computed simulation for all the test cases, and b) attend the on-line requests it receives in parallel, computing lots of similarities among the query and the stored cases to retrieve the better stored simulation.

3 Filling the Case Library: a case in the A-TEAM system

As mentioned before, the case library stores the simulation exercises too complex to be built on-line. So "*feeding*" the case library with all those simulations is an off-line

process where the content developers and the Simulation Modules managers are involved in order to create the information the CBR's Case Library will store.

While designing a course, the content developers should decide which examples will be shown, which images, the tests to be performed by the trainee and so on. Some of the examples or the tests could be emergency simulations that also have to be designed with the managers of the Simulation Modules, in order to find the proper simulation model to the simulation proposed and thus, the format of the inputs and outputs for that model.

While designing a complex emergency simulation (that is, a simulation that should be pre-computed off-line and stored by the CBR), the information that should be collected and built is composed by the parts depicted in figure 3:

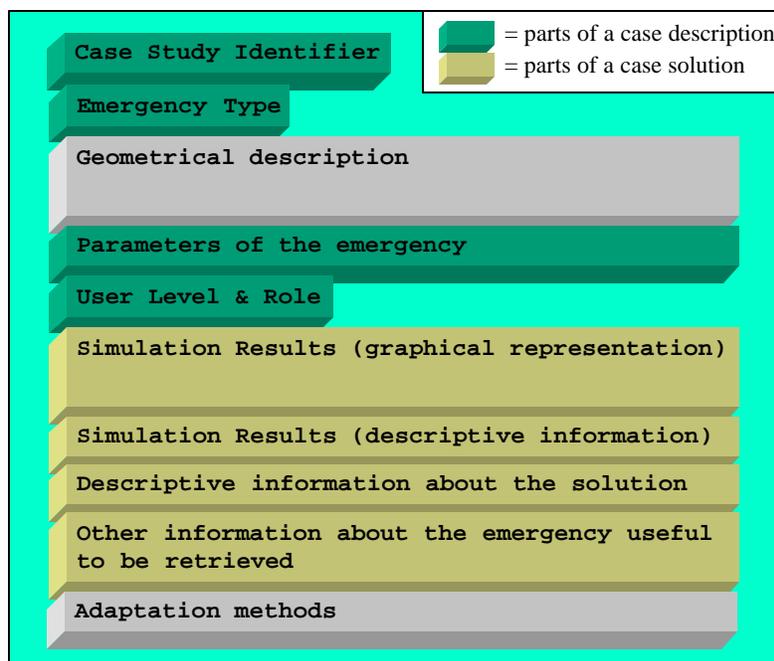


Fig. 3. Information associated to an emergency simulation.

Case Study Identifier and Emergency Type

The Case Study Identifiers are useful to distinguish among the different case studies (Austria, Italy, Switzerland, Portugal and United Kingdom), while the Emergency Types are useful to identify the kind of emergency (such as Road Tunnel, Chemical Plant, etc.). These identifiers are set for the whole A-TEAM system.

Geometrical Description

It is all the useful geometrical information about the topology (mountains, roads, buildings, etc.) of the place where the emergency will be simulated. This piece of in-

formation is not used as part of the case in the Case Library, as it's only useful for the Simulation Modules in order to run their simulation. More details about the format of these data are provided in deliverables of A-TEAM project (A-TEAM, 2003).

Parameters of the emergency

These are the parameters that set the initial state of the emergency. They are established for each type of emergency. For instance, in emergencies that involve gas dispersion, some of the parameters are Wind speed, Wind direction, Air Temperature, Exit velocity (of the dangerous substance that is spilled), etc.

These parameters are not only useful to the Simulation Modules as input of their runs, but also to the CBR as part of the case description.

User Level and Role

This piece of information is placed if needed. Useful to tag emergency exercises designed specifically for a given trainee profile (a team coordinator, a fireman, etc.) or a certain level of expertise (beginner, intermediate, expert). This is needed when the content developer decides that different user profiles (roles) or levels of expertise need different simulations to be shown.

Simulation Results

They are the result of the runs of the Simulation Modules to be stored for a later delivery to RTXPS when requested, as part of the case solution.

They consist on two kinds of data:

- The graphical information is useful to run the simulation on the trainee's display. Examples of this kind of data are 2D map representations of the emergency, gas concentration isolines, bar graphs, etc.
- The descriptive information is useful to provide the trainee with some numerical features or to be used in the RTXPS as part of the course management. Examples of this kind of information are the final values of temperature in the affected area, the concentration values of toxic substances in that area.

Descriptive information about the solution

Along with the simulation results, it could be useful to include some information (maybe only some hints) about the proper solution to solve the emergency, or the proper measures to make the area to return to a normal state (for instance, the procedures to be taken to clean the area from the toxic substances).

Other information about the emergency

This is another piece of information placed only if needed. Is up to the content developers to figure if they need more information associated with each simulation, so it should be stored with the rest of data.

Adaptation Methods

As stated before, it is unfeasible to pre-compute all the possible emergency simulations the user will need. The similarity-based retrieval of CBR systems partially ad-

dresses this issue, by retrieving not the exact simulation requested but the most similar one. But there is another step that can fit the solution given by the case retrieved to the actual simulation requested: the adaptation step.

Adaptation can be made from the best case's solution or from the joint of some of the best cases. It allows creating new solutions from the case solutions stored in the Case Library. In the A-TEAM context that means that it is not needed to have pre-computed all the possible simulations with all the possible combinations in the input parameters if there's a method to adapt the results of the most similar emergency situations to the current one requested by the RTXPS.

The adaptation step is a domain-dependent process, so that means it has to be designed for each emergency type. Depending on the kind of information to adapt it can be expressed as a mathematical functions (to compute some sort of interpolation), a piece of computer code or a list of rules to be applied.

4 Introduction of cases in the Case Library

In order to build a case library for a given case study and a given type of emergency, there are needed two kinds of information:

- The parameter descriptions: it is a metadata file used by the RTXPS system to describe each of the parameters that it uses in its reasoning process. These metadata files, which we will refer as the RTXPS parameter descriptors, are used as a standard notation in the rest of the A-TEAM system, so they are also an useful input for the CBR system.
- The Simulated Emergencies: the Simulation Modules following the specifications of the content providers provide these data. Each of the simulated emergencies would become a case in the Case Library.

In figure 4 the inputs of the CBR are depicted:

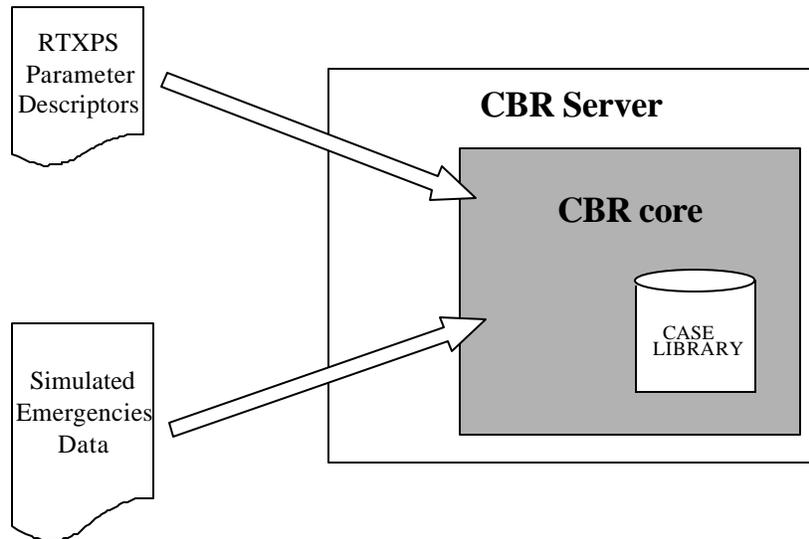


Fig. 4. The off-line inputs of the CBR server

Both inputs are files sent by e-mail or placed in a restricted ftp site to be downloaded. The format of the RTXPS parameter descriptors are describe in the A-TEAM project documentation (A-TEAM, 2003). The format of the Simulated Emergencies file depends on whether the content developer decided that different user profiles (roles) or levels of expertise need different simulations to be shown. As explained before, when there were emergency exercises designed specifically for a given kind of trainee a piece of information, named User Level & Role in figure 5, is added.

```

CASELIB

HEAD
CI <ATEAM_case_study_id>
ET <A_TEAM_emergency_type_id>
INPUTP_ORDER
P <RTXPS_param_descriptor>
P <RTXPS_param_descriptor>
...
P <RTXPS_param_descriptor>
ENDINPUTP_ORDER
TRAINEE CASE_SPECIFIED
OUTPUTP_ORDER
P <RTXPS_param_descriptor>
P <RTXPS_param_descriptor>
...
P <RTXPS_param_descriptor>

```

```

ENDOUTPUTP_ORDER
ENDHEAD

BODY
CASE
N <num_case>
INPUTP
<val1> <val2> ..... <valn>
ENDINPUTP
TRAINEE
T <ATEAM_trainee_role_id>
...
T <ATEAM_trainee_role_id>
T
<ATEAM_trainee_expertise_level>
...
T
<ATEAM_trainee_expertise_level>
ENDTRAINEE
OUTPUTP
<val1> <val2> ..... <valn>
ENDOUTPUTP
OUTPUTG
<Graphic_Output_Data>
ENDOUTPUTG
SOLUTION
<Solution_Data>
ENDSOLUTION
MOREINFO
<More_information_data>
ENDMOREINFO
ENDCASE

CASE
...
ENDCASE

...

ENDBODY
ENDCASELIB

```

Fig. 5. Example of format of the Simulated Emergencies File

As shown, the input is a tagged text file composed by a) some headers, with information regarding the whole simulations, and b) a body, where each case block holds the data of a single simulation. In order to decrease the size of the file, the relation among parameter names and values is set by defining an order in the head section. So, the

order of the parameter names (which should follow the standard names set by the RTXPS descriptor file) is the order of their respective values for each simulation.

5 Design and implementation of the CBR Server

As part of the distributed A-TEAM system, the CBR Server should attend all the requests for pre-computed simulation that the main RTXPS server makes remotely. To do this, a new component should be added to the CBR server shown previously in figure 6: the HTTP server.

So, with this addition, the CBR Server is composed by:

- The HTTP server: this component deals with the communication issues of the CBR server. It is a front-end that can manage multiple HTTP connections from one or several RTXPS servers, decoding the HTTP request to get the query and sending it to the CBR core. Then it encodes the result provided by the CBR core and sends it back to the RTXPS server.
- The CBR core: this component receives several on-line queries through the HTTP server, and it returns the best pre-computed solution for each query. The current implementation lets the core to run all the searches in parallel.

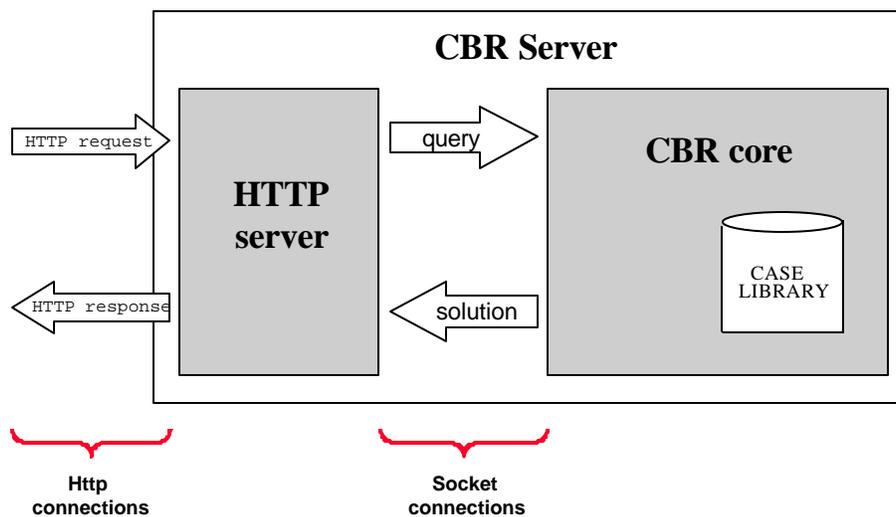


Fig. 6. The on-line inputs of the CBR Server

Java technologies have been chosen for both the HTTP server and the CBR core, not only because of its portability but also because of the multi-threading and secure networking mechanisms it provides. Therefore, the HTTP server can be any commer-

cial server that manages Java servlets, while the CBR core is a multi-threaded application.

The sequence diagram in figure 2.9 describes in detail all the fluxes of information created in an on-line query. It shows how an HTTP request is processed: the Web Server creates a new thread of execution and calls the method `doPost` of the Java Servlet (class `CBRServlet`), which, in turn, calls its `doGet` method (HTTP Get and HTTP Post requests are processed the same way by the servlet).

The `doGet` method obtains the query information, and opens a Socket connection to the CBR server (class `CBReasoner`), that is waiting for connections from the *HTTP server*. When receiving such a connection, it creates a new thread of execution (class `ServerThread`) to manage the request of information. The `ServerThread` class uses the method `importFromData` of class `ATeamCase` to create a new case from the raw data it receives through the socket, and then starts a reasoning cycle to obtain the result. The result is sent back through the socket connection to the *Java Servlet*, where the result is translated into an HTTP response format and then sent back by the *Web Server*. It is important to note that, as either the Java Servlets and the server threads use Java threads to run, the CBR server can manage more than one request concurrently.

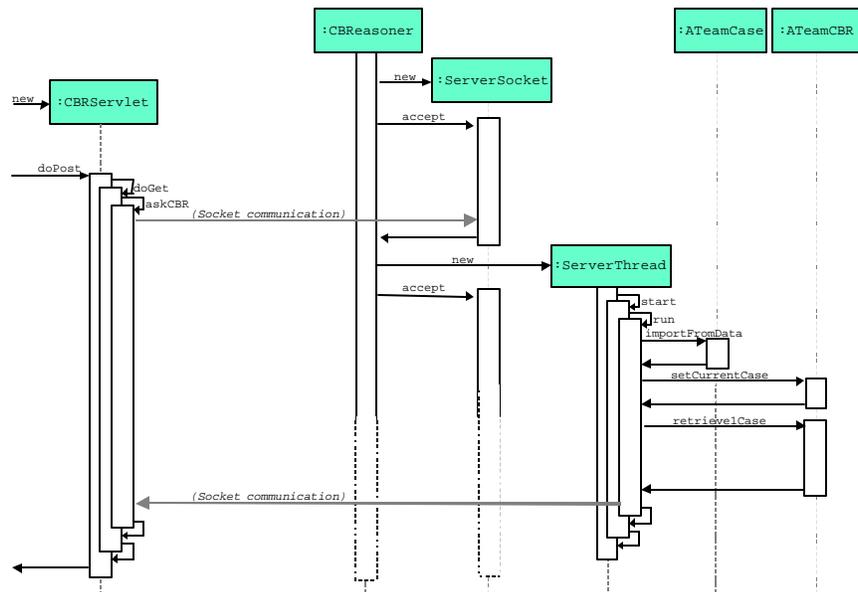


Fig. 7. Sequence diagram of a on-line query.

6 An application: a railway tunnel scenario

As an example of application, now one of the test cases of the CBR server within the A-TEAM project will be presented. The test case consists of a simulation exercise

placed in a 6.3 km. Long tunnel: the Grauholz tunnel, north of Bern, Switzerland, which is a link for east to west traffic between Zurich and Geneva, and for north to south traffic through the Alps (see figure 8).

The simulation exercise aims to present different scenarios to the student to be solved, from a train that develops a small fire in a container without dangerous goods involved to a bigger incident where a train derails inside the tunnel and releases relevant amounts of fuel oil and starts burning, heating up the tanker(s) and possibly, creating a chain reaction (rupture of the containment, thermal decomposition and formation of toxic gas) that scales up the incident. Part of the simulation exercise includes the assumption that there are passengers in the tunnel to be saved, either from the same train that started the fire or a passenger train approaching from the other direction into the tunnel which could not be stopped in time and entered the tunnel.

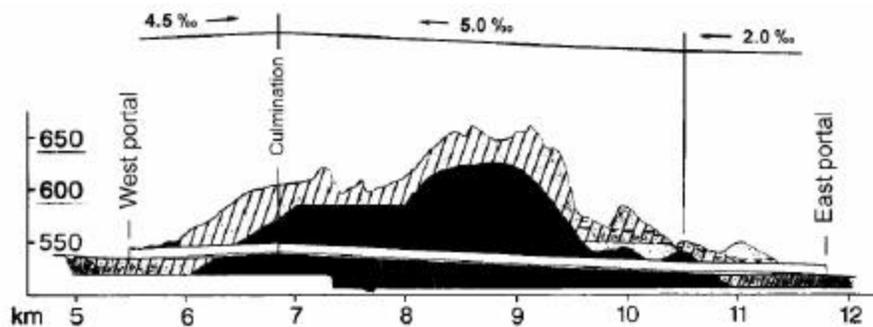


Fig. 8. The Grauholz tunnel

For each emergency case, a number of physical parameters describing the scenario characteristics were computed. For the emergency intervention and self rescue actions of people in the tunnel, the development of the following parameters during the emergency situation (fire) are crucial and constitute the output of the simulations for the whole volume of the tunnels:

- Air Temperature in the tunnel (K)
- Air pressure in the tunnel ($\text{kg}/(\text{m}^2 \text{s}^2)$)
- Air flow speed in the tunnel (m/s)
- Tracer in the tunnel (kg/m^3), is meaning the concentration of an inert pollutant (i.e. smoke).

An example of the output provided by the CBR server is depicted in figure 9. The output shows the parameters of the case plus the temperature distribution inside the tunnel (the image shows the snapshot of the tunnel 25 min after the fire started). At the bottom of the window there is a navigation tool that allows going forward/backward in the simulation time, and up/down in the tunnel (to see the temperature distribution from the ground to the ceiling of the tunnel).

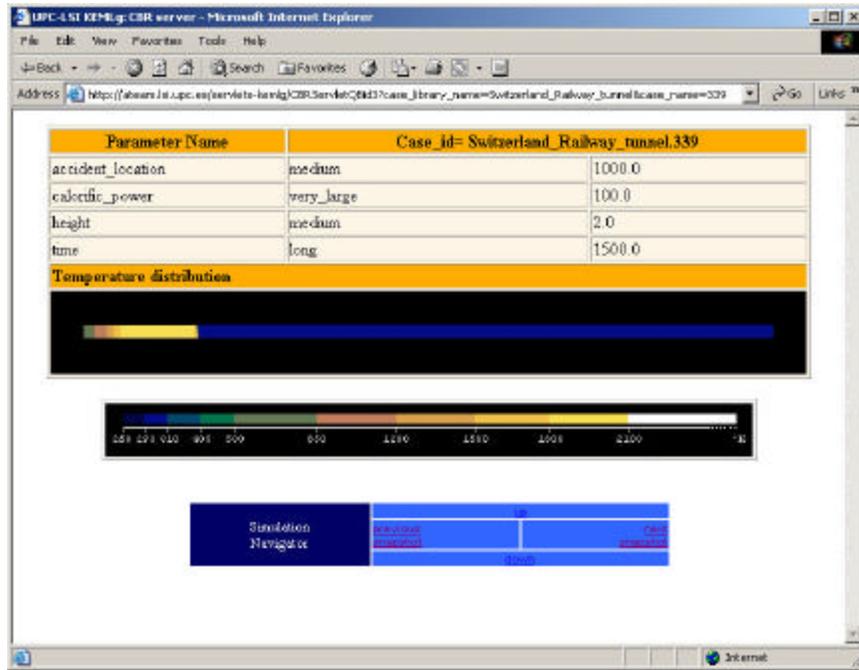


Fig. 9. Output of the CBR server for the railway scenario.

7 Conclusions and Future Work

The CBR server described within this paper has been experimentally evaluated from a qualitative point of view. The environmental manager experts who are in charge of training courses have assessed some test cases, such as the railway tunnel scenario and a road tunnel scenario.

Preliminary evaluation has shown a very good performance and accuracy in the CBR system. Only some adjustments for graphical outputs are being updated to satisfy the requirements of the users.

In the near future a more statistical and numerical evaluation will be made as the other modules and components of the A-TEAM project will be successfully integrated, and a deeper evaluation will be possible.

Also, as the design of the core of the CBR server is a very general tool it can be applied to other environmental and advanced intelligent training systems. Other domains of applications will be studied.

Acknowledgements

This work has been partially supported by the Spanish CICYT project TIC2000-1011, REC2000-1755, and EU project A-TEAM (IST 1999-10176). The authors wish to acknowledge the work and effort of all the A-TEAM partners.

References

1. A-TEAM. Advanced Training System for Emergency Management. (2003). <http://www.ess.co.at/ATEAM/>.
2. Avesani, P., Perini A., and Ricci F. Interactive Case-Based Planning for Forest Fire Management. *Applied Intelligence Journal*, Volume 13, 2000.
3. Avesani, P., Ricci, F and Perini, A. (1995) P. Avesani, F.. Combining human assessment and reasoning aids for decision making in planning forest fire fighting. *Proc. of IJCAI Workshop on Artificial Intelligence and the Environment*, pp. 71-74, Montréal.
4. Bichindaritz I. and Sullivan, K.M. Generating Practice Cases for Medical Training from a Knowledge-Based Decision-Support System. *Workshop on Case-Based Reasoning for Education and Training*, Aberdeen, Scotland. September 4th, 2002.
5. Branting, L.K., Hastings, J.D., and Lockwood, J.A. (1997) Integrating cases and models for prediction in biological systems. *AI Applications* 11(1):29-48.
6. Jones, E. and Roydhouse, A. (1995). Retrieving structured spatial information from large databases: a progress report .*Proc. of IJCAI Workshop on Artificial Intelligence and the Environment*, pp. 49-57, Montréal.
7. Krovvidy, S. and Wee, W.G. (1993). Wastewater Treatment Systems from Case-Based Reasoning. *Machine Learning* 10, pp. 341-363.
8. Ricci F., Avesani P. and Perini A. Cases on Fire: Applying CBR to Emergency Management. *New Review of Applied Expert Systems Journal*, Vol. 5, 1999.
9. Núñez H., Sánchez-Marrè M. and Cortés U. Similarity Measures in Instance-Based Reasoning. Submitted to *Artificial Intelligence*, 2003.
10. Sadek A.W., Smith B.L. and Demetsky M.J. (2001). A Prototype Case-Based Reasoning Decision Support System for Real-time Freeway Traffic Routing. *Transportation Research*, Part C, Vol. 9(5).
11. Sadek A.W., Smith B.L. and Demetsky M.J. (1999). Case-Based Reasoning for Real-time Traffic Flow Management. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 14(5).
12. Sánchez-Marrè, M., Cortés, U. R-Roda, I., Poch, M., and Lafuente, J. (1997a) Learning and Adaptation in WWTP through Case-Based Reasoning. *Microcomputers in Civil Engineering* 12(4):251-266.
13. Surma, J. and Brauschweig, B. (1996) Case-Based Retrieval in Process Engineering: Supporting Design by Reusing Flowsheets. *Engineering Applications of Artificial Intelligence*. 9(4): 385-391.