The Twofold Integration of CBR in Decision Support Systems

Paolo Avesani, Anna Perini and Francesco Ricci
I.R.S.T.
38050 Povo (TN)
Italy
e.mail: {perini}@irst.itc.it
Phone ++39+461-314330
Fax ++39+461-302040

Abstract

This short paper presents our experience in facing different levels of integration of CBR techniques with other system components and techniques inside decision support systems for controlling environmental emergencies. The integration requirement at the level of system components is distinguished from the integration requirements at the level of specific steps in the CBR problem solving cycle. We describe how these aspects have been dealt inside two systems, CHARADE and CARICA, and point out the influence of this work on our current activity.

keywords: interactive planning, constraint satisfaction problem, feature weighting.

Introduction

This paper presents our experience in using CBR techniques to develop advanced functions in decision support systems (DSSs) for complex domains, such as DSSs for controlling environmental emergencies.

In real application domains CBR can be effectively exploited when it is integrated both with other techniques and with other software system components. That is, first from a software system perspective the CBR problem solving architecture often must collaborate with other components depending on the application needs. For instance, in a DSS for controlling environmental emergencies CBR may interact with a geographical information system or with a data base or with a external environment simulator. We shall call that type of integration "collaboration".

Second, the CBR technology itself can include other techniques to address some of the steps in the CBR problem solving cycle (Aamodt & Plaza 1994). For example in the "reuse" step is becoming more and more common to use constraint satisfaction techniques to adapt the candidate solution (Purvis & Pu 1995; Wilke, Smith, & Cunningham 1998). Another example is the application of feature weighting techniques, which are borrowed from machine learning, to improve the similarity evaluation step (Wettschereck, Mohri, & Aha 1997; Ricci & Avesani 1995). This second type of integration will be called "inclusion".

We have been concerned with these two levels of integrations during the development of DSSs designed for a specific application domain: fighting forest fires. This is a complex domain due to the fact that the forest fire evolution depends on several environmental variables and can be poorly modeled, resulting in uncertainty and incompleteness of the data. Moreover the decision maker (the target user of our systems) must perform situation assessment, build a plan of first intervention, allocate and dispatch resources as soon as possible, pressed by the impending danger for people and things. A domain theory that provides a way to compute the effects of a precise decision cannot exist.

In particular we have been concerned with supporting two basic activities performed by the users of our systems: planning for the first intervention in case of an emergency; training for emergency management. Both activities rests on reasoning tasks that are usually performed by exploiting past experience in the domain, which suggests to consider CBR techniques as a common core. The effectiveness of using CBR techniques for quickly providing a draft solution to be adapted to the current situation has been pointed out also in other studies concerning crisis response planning (Gervasio, Iba, & Langley 1998). In order to develop effective functions we integrate CBR with different specialized techniques such as temporal reasoning techniques, machine learning and data visualization techniques.

These functions have been developed and tested inside two demonstrators: CHARADE, a system aimed at supporting the user in the whole process of fire fighting including both situation assessment and planning activities (Ricci *et al.* 1994), and CARICA (Avesani, Terral, & Martin 1997), a system that supports training for intervention planning.

In this paper we shall explain how we dealt with the different types of CBR integrations defined above, in the framework of these two projects.

We will consider the *collaboration* type of integration first, then the *inclusion* type.

We shall conclude the paper with one lesson. CBR can be really exploited inside a real application when available as a library of functions ready to be integrated with the other software components rather than as an independent software tool, as it is more common in commercial products. These considerations motivate our current work in developing a CBR C++ library, called CBET (Avesani, Perini, & Ricci 1997).

Collaboration

This section discusses the integration of CBR with other system components inside the CHARADE and CARICA systems.

Planning the first intervention

Planning the first intervention during a forest fire emergency, requires to mix planning and situation assessment according to a precise operational work-flow that is typical of each fire fighting organization. This activity is supported in CHARADE by a CBR planner which is integrated with a geographical information system (GIS), with a DBMS of the resources and with a numerical fire spreading model.

The relationships between the system components is depicted in Figure 1. In particular the GIS manages thematic mapping (roads, rivers, bases, etc.), visualizes the fire front evolution, as forecasted by the fire spreading model, and allows to perform spatial queries. This helps the user in focusing the spatial analysis on particular zones of the fire front (called also *sectors*) where some actions (water spraying, control line, back fire, etc.) have to be executed in order to protect people and valuable things or simply to control the fire. This spatial information (sector length and orientation, accessibility, availability of water reservoirs near the sector, type of vegetation) provides part of the predictive features used by the planner. Other predictive features concern the fire's physical parameters and are directly provided by the fire spreading model. Finally other data, which are concerned with the availability of the resources located in the bases closest to a sector, are provided by the resource data base.

On the basis of these scenario data the planner retrieves candidate plans that can be adapted to the current fire emergency. The resulting plan is a correct solution for the current emergency and defines a resource allocation problem to be solved by an additional system component, the scheduler. The solution computed by the scheduler is a list of orders to be sent to the bases for resource dispatching. This ends the intervention planning process supported by CHARADE.

Training for intervention planning

Training for intervention planning means to promote the process of learning and improving planning strategies and tactics by firemen. This kind of training activity is traditionally performed by playing a simulated emergency scenario built on the basis of past cases. Teachers of fire fighting planning define a new realistic scenario by exploring a memory of cases.

CARICA supports both the activity of feeding a case base of past emergencies, providing a specific module for cases acquisition, and the case base exploration activity, providing a second module based on CBR. The complexity of the case acquisition phase rests on the fact that the user is more familiar with a graphical representation of the plan where actions are depicted as appropriate icons on the map, as it is typical in a military domain. An example is given in Figure 2. Therefore the first module provides a specific graphical tool for this type of plan description which is a component of the case emergency description. The second module provides a range of browsing and display functions that make possible knowledge extraction from a set of cases. It allows to detect dependencies between data, acquire practical planning competencies, visualize complex data, clustering similar cases. These functions rest on the integration of CBR retrieval techniques with well rooted machine learning techniques for selecting relevant features, clustering cases and forecasting unknown values.

Inclusion

In this section we describe some examples of how specific techniques, such as constraint satisfaction and machine learning techniques, have been exploited to improve some steps inside the CBR problem solving cycle.

Constraint satisfaction for plan adaptation

The CHARADE planner, as mentioned above, retrieves plans matching scenario features resulting from the assessment phase. The candidate plan needs to be adapted to the current scenario in order to yield an executable plan. Different approaches to this task are discussed in (Veloso, Munoz-Avila, & Bergmann 1996; Leake, Kinney, & Wilson 1996), some of them (i.e. generative adaptation) are not suitable for intervention planning in the fire fighting domain that lacks a generative problem solver. In CHARADE the adaptation task is performed in two steps. The retrieved plan includes a set of constraints, on time and resource variables, that are automatically propagated and checked against other descriptive variables of the current situation. A further adaptation phase is realized through an interactive process of editing activities performed by the user, like inserting/deleting actions, modifying action durations or temporal relations between parts of the plan. The Planner architecture depicted in Figure 1 shows the use of a constraint reasoning module that manages the constraint network associated to a plan and supports both adaptation steps.

Plan adaptation exploits constraint reasoning techniques for modeling the constraints on actions and checking their consistency. In particular we focused on the temporal constraints of a plan. For example a plan for controlling the fire on a given sector can be impractical if the time required to perform it is greater than the deadline posed by the fire propagation toward that sector, actions can have minimum and maximum durations, and so on.

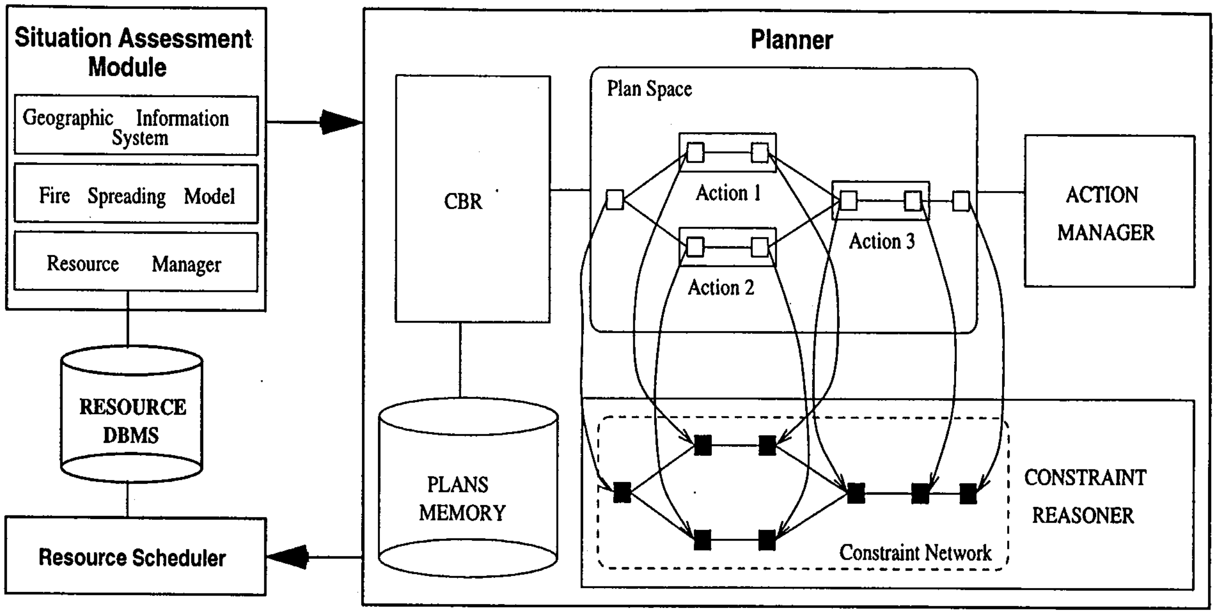

Figure 1: The components of the CHARDE system. The Situation Assessment module provides a scenario description to the CBR module. The Planner supports the user for plan definition and yields an allocation problem to be solved by the Resource Scheduler. The Planner includes: the case-based reasoner, which retrieves (stores) plans from (to) the plan memory; the plan space is the data space in which plans are installed; the action manager, that manages the user requests to modify the plans installed in the plan space; the constraint reasoner, in which variables and constraints are created and constraint reasoning algorithms called.

These constraints can be modeled in terms of bounded difference between the time points corresponding to the actions start and end times. This defines a CSP problem on variables with continuous domains according to the formalization given in (Dechter, Meiri, & Pearl 1991), the so called Temporal CSP (TCSP). The variables represent the actions start and end times. The constraints are those representing the minimum duration of actions and those representing the temporal relations between the actions. In the CHARADE system we considered a tractable subset of the TCSP formalism, the Simple Temporal Problem formalism that admits a single bounded difference constraint between two variables. Particular attention has been devoted to the analysis of how the different temporal reasoning computations that can be performed on a STP (determining the network consistency, computing the feasible times for the variables of the network, computing the minimal network representation) can be better implemented and used in order to efficiently support the interactive process (Perini & Ricci 1995). This analysis motivated our research on efficient algorithms for the incremental updating of constraint networks (Gerevini, Perini, & Ricci 1996).

Feature weighting for case retrieval

Feature weighting is a technique, which originates from pattern recognition and machine learning, that has become popular in many CBR systems. The CHARADE system uses it to select in a more appropriate way the plan applicable in a given emergency situation. Conversely, in the CARICA system, where CBR is mainly used for information retrieval purposes, weighting is strongly influenced by the users goals expressed by different queries to the system.

The retrieval function in the CHARADE planner exploits a novel approach to compute nearest neighbor based on a local metric that we called LASM (local asymmetric similarity metric) (Ricci & Avesani 1996). This approach rests on two basic assumptions. The first one (locality) states that the metric is defined locally: the space around a trial case is measured using the metric attached to that case. The second one (asymmetric) states that the distance between two points in a continuous feature space F_i is not symmetric, i.e., $d_i(x_i, y_i) \neq d_i(y_i, x_i)$. In fact we use two different weights for the "left" and the "right" directions.

We also provide a learning procedure for adapting the local weights to the input space. This procedure is a typical feedback feature weighting method (Wettschereck, Mohri, & Aha 1997). Our model basically implements an anytime algorithm (Boddy 1991) that updates the distance between an input case c and its neighbors depending on the role of the neighbors in solving c. If the nearest neighbor nn can be used correctly to solve c the distance between c and nn is decreased (reinforcement), otherwise the distance between c and nn is increased (punishment). These tech-

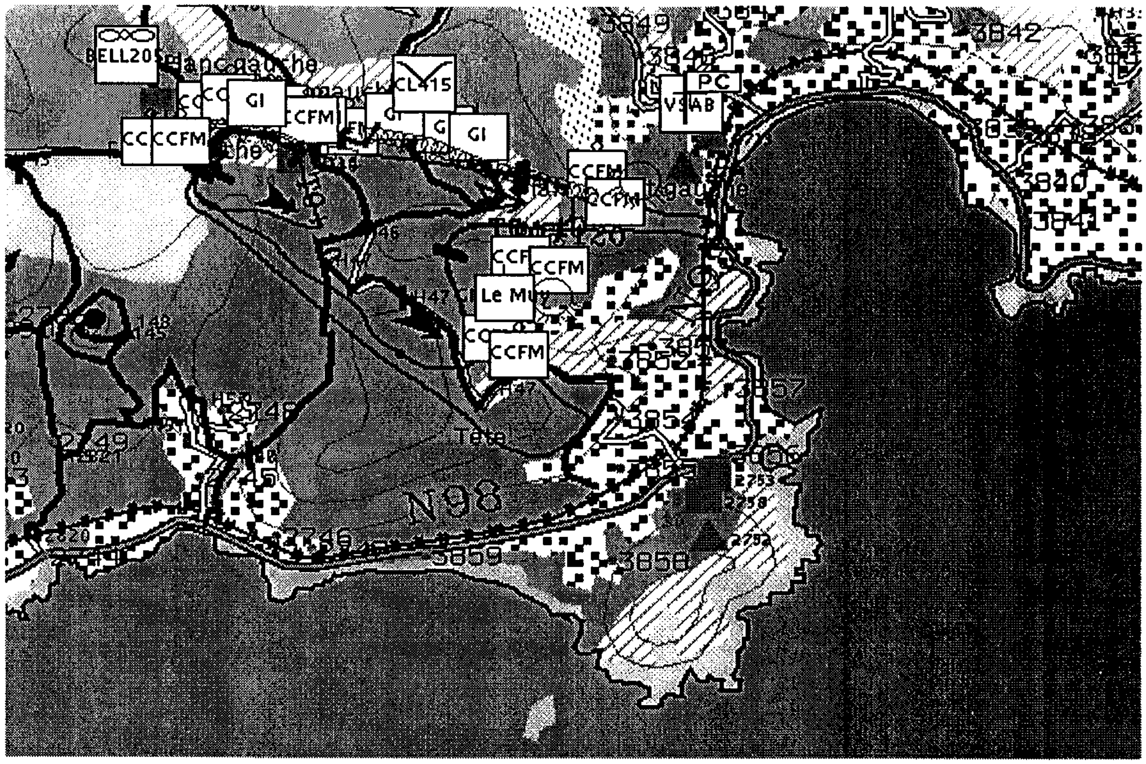

Figure 2: Example of the graphical plan representation used by firemen. The fire front evolution at different times, are drawn as closed lines. The black arrows indicate the wind direction. The labeled box icons represents the different actions to be performed (in the zone where the icon is located). For instance the icon labeled *CCFM* indicates the action of water spraying with a medium tank truck, at the fire head.

niques were fully evaluated through a set of experiments on popular datasets, as described in (Ricci & Avesani 1995). Concerning the fire fighting domain the validation activity performed by the final user focused mainly on the higher level functions of the system so that the effectiveness of these retrieval techniques for the specific domain was only partially tested.

In CARICA retrieval has not been primarily designed for classification or problem solving, even if that can be supported. Retrieval must satisfy user's information needs when a query is executed by the system and feature weighting is therefore tailored for that purpose. In a second application, when the user asks the system for a graphical plot of a given feature the system weights those features that explain better the variation of the input feature in the dataset (Chambers & Cleveland 1983).

For these purposes we have implemented in CARICA weighting methods based on Information Theory. If the user is *focused* on a particular feature f_T, and he/she is interested in discovering what other features may account for the behavior of f_T, then a statistical criterion may be used to detect such a dependency. We hypothesize that the features that maximize the "purity" objective functions, which are used for top down induction of decision trees (TDIDT) (Weiss & Kulikowski 1991), can be exploited.

Conclusions

This short paper describes our experience in dealing with different integration levels of CBR techniques inside DSSs designed for supporting emergency management tasks, such as planning the first intervention (CHARADE) and training for intervention planning (CARICA).

In particular we have distinguished a first type of integration, that we have called *collaboration*, which concerns the integration of the CBR module with other software system components. At this level the CBR planner in CHARADE *collaborates* with a GIS, a data base of resources, a fire spreading model and a scheduler. The CBR exploration module of CARICA *collaborates* with a module for case acquisition built upon a specialized graphical tool for plan description. A second level of integration, called *inclusion*, concerns the exploitation of specialized techniques inside the CBR problem solving cycle, such as constraint satisfaction techniques for plan adaptation and machine learning feature weighting techniques to improve the similarity evaluation step and obtain a more effective case retrieval.

This experience stimulated a current project aimed at recasting the techniques described in this paper in a C++ library, called CBET, allowing a flexible multi-level integration of CBR components into different, specific real applications (Avesani, Perini, & Ricci 1997).

References

Aamodt, A., and Plaza, E. 1994. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications* 7(1):39–59.

Avesani, P.; Perini, A.; and Ricci, F. 1997. Cbet: a case base exploration tool. In Springer-Verlag., ed., *Fifth Congress of the Italian Association for Artificial Intelligence (AI*IA 97)*.

Avesani, P.; Terral, S.; and Martin, E. 1997. Final report. carica deliverable d11. Technical report, IRST.

Boddy, M. 1991. Anytime problem solving using dynamic programming. In *Proceedings of the Nineth National Conference on Artificial Intelligence*, 738–743.

Chambers, J. M., and Cleveland, W. S. 1983. *Graphical methods for data analysis*. Wadsworth.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49.

Gerevini, A.; Perini, A.; and Ricci, F. 1996. Incremental algorithms for managing temporal constraints. In *Proceedings of the Eighth IEEE International Conference on Tools with Artificial Intelligence, TOULOUSE, France, November 16-19*.

Gervasio, M.; Iba, W.; and Langley, P. 1998. Case-based seeding for an interactive crisis response assistant. In *Proceeding AAAI-98 Workshop on Case-Based Reasoning Integrations*.

Leake, D.; Kinney, A.; and Wilson, D. 1996. Acquiring case adaptation knowledge: A hybrid approach. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA*.

Perini, A., and Ricci, F. 1995. An interactive planning architecture. In *Third European workshop on Planning, Assisi, Italy September 27-29*. T"New Directions in AI Planning", IOS Press.

Purvis, L., and Pu, P. 1995. Adaptation using constraint satisfaction techniques. In *International Conference on Case-Based Reasoning (ICCBR-95), Sesimbra, Portugal, Oct. 23-26*. Spinger Verlag.

Ricci, F., and Avesani, P. 1995. Learning a local similarity metric for case-based reasoning. In *International Conference on Case-Based Reasoning (ICCBR-95), Sesimbra, Portugal, Oct. 23-26*.

Ricci, F., and Avesani, P. 1996. Exact learning and data compression with a local asymmetrically weighted metric. In *ICML-96 Workshop on Learning in Context-Sensitive Domains, Bari, July 3rd,*.

Ricci, F.; Mam, S.; Marti, P.; Normand, V.; and Olmo, P. 1994. CHARADE: a platform for emergencies management systems. Technical Report 9404-07, IRST.

Veloso, M. M.; Munoz-Avila, H.; and Bergmann, R. 1996. General-purpose case-based planning: Methods and systems. *AI Communications*.

Weiss, S. M., and Kulikowski, C. A. 1991. *Computer Systems that Learn*. Morgan Kaufmann.

Wettschereck, D.; Mohri, T.; and Aha, D. W. 1997. A review and comparative evaluation of feature weighting methods for lazy learning algorithms. *AI Review Journal* 11:273–314.

Wilke, W.; Smith, B.; and Cunningham, P. 1998. *Case-Based Reasoning Technology from Foundations to Applications*. Springer-Verlag. chapter Using configuration techniques for adaptation.

CBR Integrations Questions

1. **Integration name/category**: CHARADE
2. **Performance Task**: Situation assessment & planning
3. **Integration Objective**: Plan adaptation
4. **Reasoning Components**: Constraint reasoning (temporal)
5. **Control Architecture**: CBR as master
6. **CBR Cycle Step(s) Supported**: Reuse, Revision
7. **Representations**: Simple temporal constraints
8. **Additional Reasoning Components**: GIS, DBMS, scheduler, user (adaptation)
9. **Integration Status**: Applied
10. **Priority future work**: Integration with decision-theoretic planning techniques

1. **Integration name/category**: CARICA
2. **Performance Task**: Training device for intervention planning
3. **Integration Objective**: Information Retrieval
4. **Reasoning Components**: Inductive feature weighting
5. **Control Architecture**: CBR as master
6. **CBR Cycle Step(s) Supported**: Retrieval
7. **Representations**: User information needs
8. **Additional Reasoning Components**: Browser, data visualization
9. **Integration Status**: Applied
10. **Priority future work**: Porting and engineering