This is a preprint of an article published in Proceedings of 13th European Conference on Artificial Intelligence, ed. Henri Prade, John Wiley, pp58-62, August 1998.

Hierarchical CBR for Multiple Aircraft Conflict Resolution in Air Traffic Control[1]

Andrea Bonzano and **Pádraig Cunningham**[2]

Abstract. In this paper we present a Case-Based Reasoning system that helps air traffic controllers to solve aircraft conflicts. In particular, we focus on the hierarchical aspect of the CBR system which is able to solve multiple aircraft conflicts, i.e. conflicts that involve three or more aircraft. It is not practical to build a case-base for the different multiple aircraft possibilities as has been done for two aircraft conflicts. Instead we explore the possibility of using case fragments from two aircraft conflicts in multiple aircraft situations. The hierarchical structure that we describe here makes this possible. This involves the use of some high-level analysis of the solutions coming from the case base because the solution to a multiple aircraft conflict is not necessarily one of the solutions of the component two aircraft conflicts. The hierarchical structure allows the use of the same case-base for both two aircraft conflicts and multiple aircraft conflicts with big savings in space and time.

1 INTRODUCTION

Case-Based Reasoning (CBR) has emerged from research in cognitive psychology as a model of human memory and remembering. It has been embraced by researchers of AI applications as a methodology that avoids some of the knowledge acquisition and reasoning problems that occur with other methods for developing knowledge-based systems. It is intuitively appealing because people readily accept that much of human competence is based on the reuse of solutions to previously encountered problems. However the reality of CBR falls short of the flexibility of reuse that humans can achieve. The standard CBR structure provides a restricted form of reuse where the new problem needs to be isomorphic to a problem in the case-base.

In this paper we explore how a hierarchical approach to CBR [1-5] can allow for more flexible case reuse. We describe a hierarchical extension to ISAC (Intelligent System for Aircraft Conflict Resolution), a CBR system that helps air traffic controllers to solve conflicts between sets of aircraft [6]. ISAC assists the controllers in the first two stages of the conflict resolution process: the selection of the aircraft to manoeuvre and decision of the type of manoeuvre. The third stage, the specification of the details of the manoeuvre, is left to the controller. While it has been possible to build a case that provides adequate cover for two-aircraft conflicts (TACs) it is not practical to provide dedicated cases to cover the different types of multiple aircraft conflicts that can arise. For instance, the third aircraft in a three-aircraft conflict increases the number of possibilities by an order of magnitude.

The hierarchical structure that we describe here allows for the reuse of TAC case components in solving MACs. This greatly extends the reusability of cases and makes the solving of MACs practicable using a case-based approach. Before describing the hierarchical extensions to ISAC we present an overview of air traffic control and multiple aircraft conflicts in section 2. In section 3 we describe the case representation used in ISAC and in section 4 we present the possibilities for hierarchical CBR in air traffic control. In section 5 we work through and example of how ISAC can solve a three aircraft conflict and present the results of a preliminary evaluation of the system before concluding in section 6.

2 MULTIPLE AIRCRAFT CONFLICTS

Internationally agreed rules exist defining separation standards below which two aircraft are said to be "in conflict". Minimum horizontal separations are typically 5 nautical miles (1nm = 1852m) in radar controlled regions and either 1000 ft or 2000 ft vertically, depending on altitude. In areas not covered by the radar the horizontal separation is bigger. In practice, controllers will often apply separations significantly larger than 5 nautical miles or 1000 ft, mainly due to the difficulties they have in accurately visualising future trajectories and conflict situations.

In [7], three types of conflict sets have been identified as being the most common:

- one versus one: the two conflicting aircraft are isolated from other conflicts;
- one versus two: two separated conflicts sharing a common aircraft;
- three-at-once: three conflicts among three aircraft.

Two Aircraft Conflicts (TACs) are already efficiently solved by the non-hierarchical version of ISAC [6]. The performance of this version could be increased with an improvement of the case-base.

In a Multiple Aircraft Conflict (MAC) the situation is more complex. If a MAC is decomposed into TACs, there is the risk of solving the wrong pair first. An overall view is necessary to decide which aircraft has to be manoeuvred.

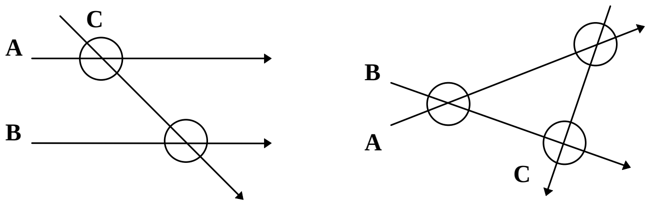


Figure 1: Types of Multiple Aircraft Conflicts.

[1] This research has been funded by Eurocontrol Experimental Centre, Paris
[2] Dept. of Computer Science, Trinity College, Dublin, Ireland.

© Andrea Bonzano and Pádraig Cunningham
ECAI 98. *13th European Conference on Artificial Intelligence*
Edited by Henri Prade.
Published 1998 by John Wiley and Sons Ltd.

As shown in Figure 1, a MAC involving *n* aircraft can be of two types: simple MAC and complex MAC. In a simple MAC all the *n-1* conflicts are generated by the same aircraft. On the other hand, in a complex MAC the conflicts are generated by different aircraft and there are at least *n* conflicts.

The different approaches to conflict resolution typical of each controller become even more evident when the conflict is a MAC. Some controllers consider only the complex MAC to be a "real" MAC. A simple MAC is only seen as a succession of TACs which are more or less interdependent.

The need for solving multiple aircraft conflicts influenced two aspects of the construction of ISAC: a case representation that takes into account MACs and a hierarchical structure.

3 GRANULARITY OF REPRESENTATION

A key design criterion in ISAC has been the development of a case representation that will be extendible from two aircraft conflicts to conflicts involving three or more aircraft. This militates against having a single conflict as the basic unit of retrieval. Conflicts should be decomposable so that the basic unit of retrieval is an individual aircraft.

The problem of representing cases describing two conflicting entities has already been faced in the CBR literature, for example in two classical systems, Mediator [8] and Persuader [9], and more recently in Truth-Teller [10]. In all these systems, perhaps because they describe interaction between humans, there is a vocabulary to characterise the "type" of conflict and this is critical in determining the solution. This is less true in ATC where the solutions depend on the arrangement of the aircraft and the context of the individual aircraft as described by their flight plans. The conflict between two aircraft can be described roughly with one or two global parameters but the final solution depends on a lot of dependent variables related to a single aircraft. For this reason the approach adopted in ISAC is somewhat different to the above systems, with an emphasis placed on some parameters that describe an aircraft on its own [6].

Two different case representations have been introduced. The first option was to create one case for each conflict, with the description of both the aircraft in the same case. This option has been referred to as "TwoInOne", because two aircraft are in one case (see Table 1).

This suggests a hierarchical CBR structure [2] where problems are represented by multiple cases. This has the big advantage that the number of aircraft that can be involved in a conflict is not limited to two. However it is more difficult to come up with a set of parameters that can capture all the details. The problem with the "OneInOne" case representation is that, by splitting the conflict into two separate cases, there is the risk of loss of information.

4 HIERARCHICAL STRUCTURE

The straightforward approach to the solution of MACs would be the creation of a new case-base containing multiple aircraft conflicts. This approach cannot be easily implemented because a MAC can involve 3, 4 or more aircraft and it is not possible to build a coherent structure for each possibility. Moreover, since a well covered case-base for TACs is already very big, the case-base for MACs would be larger still, making it impossible to build it in reality.

Table 1. A case expressed in the "TwoInOne" case representation.

Casename	Case690
HorConflConf	crossing
Priority	same
AltitudeNow	same
Speed	faster
AltConfiguration(A)	stable
CloseToTOD(A)	155
CloseToBoundaries(A)	4.8
Manoeuvrability(A)	.78
EasyToExitHorizontally(A)	easy
LevelsAvailable(A)	yes
Faster(A)	difficult
Slower(A)	difficult
AltConfiguration(B)	stable
CloseToTOD(B)	352
CloseToBoundaries(B)	8.3
Manoeuvrability(B)	.78
EasyToExitHorizontally(B)	possible
LevelsAvailable(B)	yes
Faster(B)	difficult
Slower(B)	difficult
Solution	*dow1*

The second option, referred to as "OneInOne", was to create two separate cases for each conflict, one from the perspective of each aircraft (see Table 2). Storing the description of the two conflicting aircraft in the same case is the most obvious choice because it reflects the controller's way of examining a conflict, but it is not easily extendible to multiple aircraft conflicts. In the "OneInOne" conflict representation the information about the other aircraft involved is implicit in the environment description.

Table 2. Two cases expressed in the "OneInOne" case representation.

Casename	Case690(A)	Casename	Case690(B)
HorConflConf	crossing	HorConflConf	crossing
AltitudeNow	same	AltitudeNow	same
AltConfiguration	stable	AltConfiguration	stable
Speed	faster	Speed	slower
CloseToTOD	155	CloseToTOD	352
CloseToBoundaries	4.8	CloseToBoundaries	8.3
Manoeuvrability	.78	Manoeuvrability	.78
Priority	same	Priority	same
EasyToExitHoriz.	easy	EasyToExitHoriz.	possible
LevelsAvailable	yes	LevelsAvailable	yes
Faster	difficult	Faster	difficult
Slower	difficult	Slower	difficult
Solution	*dow1*	*Solution*	*dow2*

An alternative to the straightforward approach is a hierarchical structure. With a multiple aircraft conflict a hierarchical structure of the same type as the system Déja-Vu [11] can be used. We reuse the case-base of the two aircraft conflict by building an abstraction hierarchy as done in CADET [12].

Three hierarchical structures for the solution of the MAC are suggested: Independent CBR, Look ahead CBR and Hierarchical

CBR. In evaluating these alternatives it transpires that solutions valid for a simple MAC are valid also for a complex MAC.

4.1 Independent CBR

Let us suppose that the 3 aircraft A, B and C are involved in the MAC A-B-C composed of the two TACs A-B and A-C. With the "OneInOne" case representation an independent case is created for each aircraft involved in the conflict. No track is kept of the two TACs A-B and A-C because the conflicts are represented with no-go zones for each aircraft.

This means that the MAC is not decomposed into TACs. ISAC solves the conflict for each of the aircraft involved in the MAC. The solutions found for each aircraft are then examined and a common solution for the MAC is extracted. This structure for a complex MAC with aircraft A-B-C is shown in Figure 2 and the name "Independent CBR" comes from the fact that the aircraft are described in independent cases.

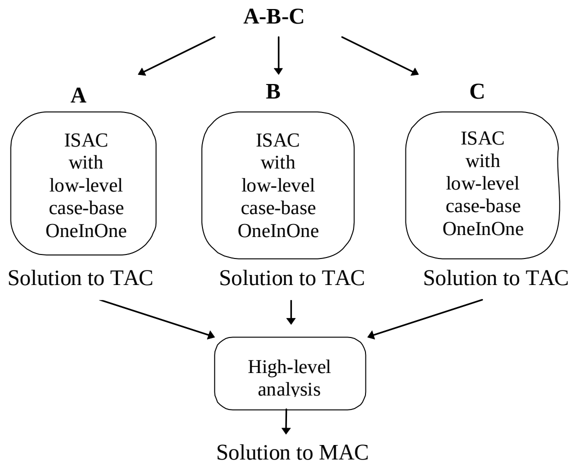

A-B-C
A
B
C
ISAC with low-level case-base OneInOne
ISAC with low-level case-base OneInOne
ISAC with low-level case-base OneInOne
Solution to TAC
Solution to TAC
Solution to TAC
High-level analysis
Solution to MAC

Figure 2: Independent CBR

4.2 Look ahead CBR

With this structure, the simple MAC A-B-C is decomposed into the two TACs A-B and A-C (or three TACs, AB, AC, BC for a complex MAC) which are solved separately by the system either with the "TwoInOne" or with the "OneInOne" case representation. Some heuristic rules are necessary to combine the solutions to the TACs into a coherent solution for the MAC. It should be noted that in this structure the "OneInOne" case representation is used to solve the TACs separately, whereas in the Independent CBR structure the same case representation is immediately used to solve the MAC conflict. Figure 3 shows how the Look Ahead CBR structure works.

4.3 Hierarchical CBR

This structure is the most abstract and the one that needs the biggest changes to the original structure of ISAC. The MAC A-B-C is examined at a high level to see if it is possible to immediately find a solution. A new high level case-base must be introduced for this first step. If no immediate solution is found, the high level case-base introduces some constraints or new parameters that are then used in the next step where the low level case-bases for the TACs are used. Again, the solutions found for the TACs have to be filtered to give a coherent general solution.

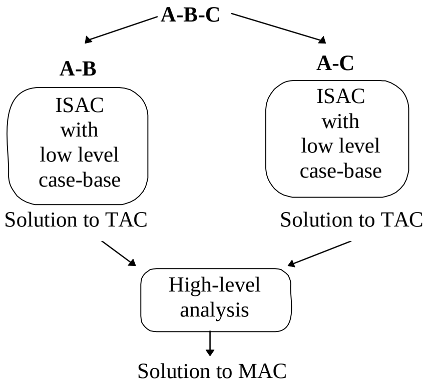


Figure 3: Look ahead CBR

One of the disadvantages of using this genuinely hierarchical approach is that a high level case-base becomes necessary and this case-base has to be built from scratch. Some of the parameters that might be used in the high level case-base are:

- geometrical description of the conflict (vertical view).
- geometrical description of the conflict (horizontal view).
- Is there an aircraft common to all the conflicts? (i.e. is it a simple MAC or a complex MAC?).
- If yes, some data about the aircraft which is in conflict with all the others.

The output that the case-base will give, as seen in Figure 4, is either the solution to the MAC or some extra constraints that can be used by the low level case-bases. (As stated before, this structure would work equally well for a complex MAC made up of three or more TACs)

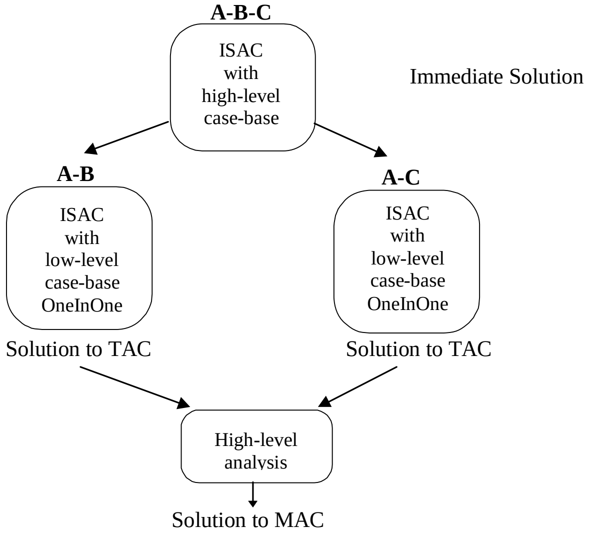

A-B-C
ISAC with high-level case-base
Immediate Solution
A-B
A-C
ISAC with low-level case-base OneInOne
ISAC with low-level case-base OneInOne
Solution to TAC
Solution to TAC
High-level analysis
Solution to MAC

Figure 4: Hierarchical CBR

4.4 High-level analysis

In Figures 2, 3 and 4, the last step before the final solution has been named "high level analysis". This analysis is necessary to extract a coherent global solution from the solutions to the simple TACs. An example of a "cheap" analysis is to choose the solution of the TAC that has been retrieved with the highest activation as the solution for the MAC. In this situation the drawback is that the general view of the conflict is not taken into account.

Another example of analysis is that used in a rule-based system called AIRPAC which chooses the first conflict in order of time and applies that solution. AIRPAC first looks for a rule able to solve all the conflicts in a co-ordinated way. If it does not find anything, it decomposes the conflict and the sub-conflicts are solved [7]. Even if in this situation the searching algorithm would be faster because only the solution for the first TAC is necessary, this analysis proves too myopic: solving the first conflict in time is not necessarily the best global solution. In the latter option the high-level analysis comes before an effective search because the first conflict must be chosen. A similar structure occurs in the Hierarchical CBR where the high-level case-base could be replaced by a set of rules that perform the same analysis.

Having all the solutions to the conflicts available, on the other hand, even if more time consuming, gives a broader view of the conflict and thus the high-level analysis can be more general. It is important to note that two different types of case reuse have been mentioned: case reuse with a hierarchical structure and case reuse with the "OneInOne" case representation. Those are two different approaches to the same problem. The hierarchical structure reuses two aircraft conflicts for solving multiple aircraft conflicts, without altering the structure of the TAC cases. A different approach is to explicitly change the case representation with the purpose of reusing each single aircraft description in any type of conflict.

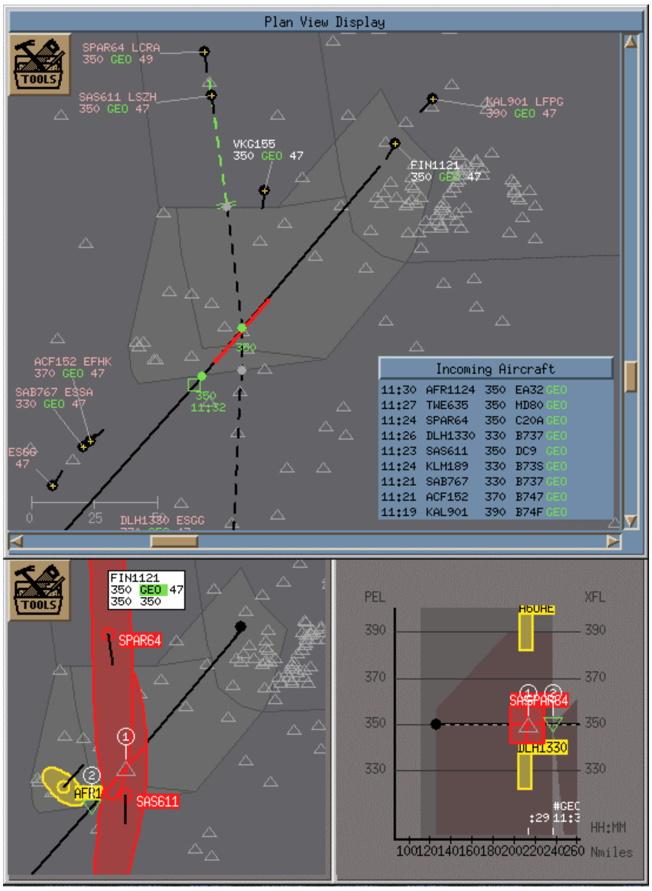
5 AN EXAMPLE

Of the three options for the resolution of a multiple aircraft conflict, the only one that has been implemented so far in ISAC is a simplified version of the "Look Ahead CBR". The "Independent CBR" option has been discarded because the performance of the system when using the "OneInOne" case representation was not as good as the performance with the "TwoInOne" case representation. The "Hierarchical CBR" option has not been implemented yet for reasons of time: the construction of a new case-base implies finding from scratch new parameters and new cases to fill the new case-base.

The heuristic rules used for the high level analysis have been suggested by controllers and could change depending on the hierarchical structure used and on the controller's preferences. For the "Look Ahead CBR", these heuristic rules are:

- check if, among the solutions to the TACs, there is a solution common to all the TACs. If yes, this common solution becomes the solution to the MAC.
- If no common solution is found, an aircraft manoeuvred in all the TACs is searched for. If found, the solution valid for that aircraft is given as solution to the MAC.
- If no common aircraft is found, the TAC closest in time is solved and that becomes the solution of the MAC.

We now describe how the Look Ahead structure works when applied to a real MAC, shown in Figure 5. In this conflict, the aircraft FIN1121 (coming from the north east) is crossing the trajectory of the two aircraft SAS611 and SPAR64 (coming from the north). At the same time, the aircraft SPAR64, behind, is catching the SAS611, which is in front and slower. All the three aircraft are flying at the same level.



Incoming Aircraft

```
11:30 AFR1124 350 EA32 GEO
11:27 TWE635  350 MD80 GEO
11:24 SPAR64  350 C20A GEO
11:26 DLH1330 330 B737 GEO
11:23 SAS611  350 DC9  GEO
11:24 KLM189  330 B73S GEO
11:21 SAB767  330 B737 GEO
11:21 ACF152  370 B747 GEO
11:19 KAL901  390 B74F GEO
```

Figure 5: A multiple aircraft conflict

The first step of the Look Ahead structure involves the resolution of the 3 constituent TACs: SAS611-SPAR64, FIN112-SAS611 and FIN1121-SPAR64. The solutions found by ISAC for the three conflicts are, respectively, "lock the speed of SAS611 and SPAR64", "climb FIN1121" and "climb "FIN1121".

The second step of the Look Ahead algorithm consists of a high-level analysis of the three TACs solutions found and the extraction of a coherent one. Because there are three TACs, at least two solutions have to be extracted. Because the solutions for the two TACs FIN1121-SAS611 and FIN1121-SPAR64 are the same, this will be the final solution for the MACs together with the solution for the SAS611-SPAR64 conflict. The Look Ahead structure for this MAC is shown in Figure 6.

If no solution in common to all the TACs was found, ISAC would have suggested the solution to the TAC closest in time as solution for the MAC. As already said, the high-level analysis could be refined with the introduction of either more rules or a high-level case-base containing more general parameters. For example, a rule stating that the aircraft which is in conflict with all the others should be moved, could be added.

The conflict used in this example is a complex MAC because there are 3 aircraft involved in 3 conflicts. As already said, the three hierarchical structures work for both simple and complex MACs.

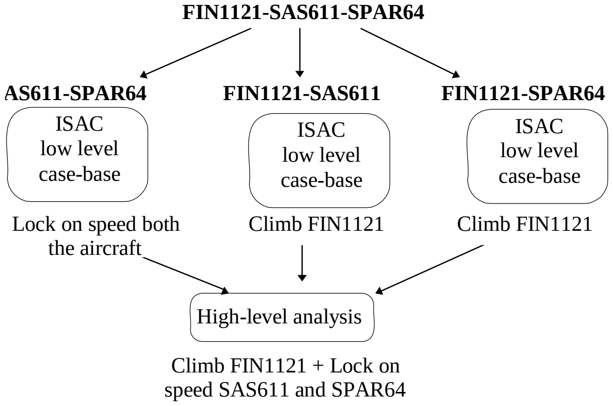


Figure 6: Look Ahead CBR for the sample MAC

The hierarchical structure of ISAC has been evaluated on 10 MACs and it gives the correct solution in all the conflicts. This result is not reliable because the case-base was known to include cases similar to these 10 targets. This limited evaluation shows that the hierarchical mechanism works and allows TAC case components to be used in MACs. This extends the reusability of cases which is the main message of this paper.

A proper evaluation of the hierarchical extensions to ISAC awaits the availability of air traffic samples with significant numbers of realistic MACs.

6 CONCLUSIONS

In this paper we present the problem of solving multiple aircraft conflicts in air traffic control. Even though human competence in air traffic control is experience based this problem is not directly amenable to a solution using CBR because of the size of the case-base required. We have shown how this problem can be addressed using a hierarchical approach to CBR.

We have analysed two general approaches to solving multiple aircraft conflicts. One approach involves the use of a case representation that contains one aircraft in each case. This case representation has been called "OneInOne". This method has not been successful because, by describing the aircraft independently of the conflict, the essential general information concerning the conflict is lost. The other general approach that implemented a hierarchical CBR structure is more effective and two alternatives have been presented.

As described by Leake [13], the most important component of a CBR system is its library of cases. This was particularly true for ISAC. First, the absence of an adaptation mechanism made it necessary to have a case-base with good coverage. Second, the complexity of the domain implied that the case-base contained lots of cases. Moreover, different controllers can give different solutions to the same conflict, mainly because they are influenced by their background.

Obviously, the most immediate necessity for ISAC is the construction of an effective case-base: big enough and with coherent solutions. Then, more work on the hierarchical structure for multiple aircraft conflicts will be done. The rules used in the high level analysis could be substituted with a small and more general case-base especially conceived for MACs. The parameters used in this case-base would be different, even if still related, to the ones used in the case-base for TACs.

REFERENCES

[1] M.A. Redmond, Distributed Cases for Case-Based Reasoning: Facilitating the Use of Multiple Cases, *Proceedings of the 8th National Conference on Artificial Intelligence*, Boston, USA, pp. 304-309, 1990.

[2] B. Smyth, P. Cunningham, Déjà-Vu: A Hierarchical Case-Based Reasoning System for Software Design, *Proceedings of 10th European Conference on Artificial Intelligence,* Vienna, B. Neumann Ed., Wiley & Son, pp.587-589, 1992.

[3] Maher, M.L., & Zhang,D.M., "CADSYN: A Case-Based Design Process Model", *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, Vol 7, No. 2, pp. 97-110, 1993.

[4] Bergmann, R., & Wilke, W., Building and Refining Abstract Planning Cases by Change of Representation Language, *Journal of Artificial Intelligence Research,* Vol. 3, pp. 53-118, 1995.

[5] Branting, L.K., & Aha, D.W., Stratified Case-Based Reasoning: reusing Hierarchical Problem Solving Episodes, *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada, pp. 384-390, 1995.

[6] Bonzano A., Cunningham P., Meckiff C., ISAC: A CBR System for Decision support in Air Traffic Control, Advances in Case-Based Reasoning, *Proceedings of the 1996 European Workshop on Case-Based Reasoning*, I. Smith and B. Faltings Eds., Springer Verlag Lecture Notes in Artificial Intelligence, pp.44-57, 1996.

[7] Shively C., Schwamb K.B., AIRPAC: Advisor for the Intelligent Resolution of Predicted Aircraft Conflicts, *Mitre Corporation*, MTR-84W164, October 1994.

[8] Simpson R.L. , A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute mediation, *Ph.D. Thesis, Georgia Institute of Technology*, School of Information and Computer Science, GIT-ICS-85/18, 1985.

[9] Sycara E.P. Resolving Adversarial Conflicts: An Approach to Integrating Case-Based and Analytic Methods, *Ph.D. Thesis, Georgia Institute of Technology,* GIT-ICS-87/26, 1987.

[10] Ashley K.D., McLaren B.M., Reasoning with Reasons in Case-based Comparisons, *Proceedings of the 1995 International Conference on Case-Based Reasoning, Case-based Reasoning Research and Development*, Lecture Notes in Artificial Intelligence, M. Veloso and A. Aamodt Eds., Springer Verlag, pp.133-144, 1995.

[11] Smyth B., Cunningham P., Complexity of Adaptation in Real-World Case-Based Systems, *Proceedings of the 6th Irish Conference on Artificial Intelligence and Cognitive Science*, R. Cowie and M. Owens eds., pp.229-240, 1993.

[12] Sycara E.P., Navinchandra D., A Process Model of Experience-Based Design, *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society.* 1989.

[13] Leake D.B. *Case-Based Reasoning: Experiences, Lessons and Future Directions, Chapter 2,* AAAI Press, The MIT Press, 1996.