ISAC: A CBR System for Decision Support in Air Traffic Control[1]

Andrea Bonzano* **, Pádraig Cunningham*, Colin Meckiff**

*Reasoning Research Project
Department of Computer Science
Trinity College Dublin
College Green
Dublin 2
Ireland
{Andrea.Bonzano,Padraig.Cunningham}@cs.tcd.ie

**Eurocontrol Experimental Centre
91222 Brétigny Sur Orge
France
Colin.Meckiff@eurocontrol.fr

Abstract. The conflict resolution task performed by air-traffic controllers appears a suitable task for automation using CBR. This is because human competence seems to involve recognising situations and reusing solutions. In this paper we present our experiences in developing a CBR system to support this conflict resolution task. We discuss the problems of case representation: the macro problem of what should constitute a case and the micro problem of how to characterise a case. We evaluate some alternative case representations and identify a representation with one aircraft per case that is extendible to describe conflicts with multiple aircraft.

1. Introduction

In this paper we describe the application of CBR to a real world problem in air traffic control. This application is interesting in itself because of the safety critical issues involved. It is also of particular interest from a CBR viewpoint because of the question of what should constitute a case in this problem domain. At this stage in the progress of CBR research this question is chestnut that has exercised many researchers. CBR is intuitively appealing because it manifestly reflects one of the important ways in which humans solve problems. However the represent-retrieve-reuse model of CBR is often difficult to apply in situations where human competence is obviously reuse-based. This difficulty is almost always associated with the granularity of retrieval and the question of what should constitute a case.

ISAC is a case-based decision support system for conflict resolution in air traffic control (ATC). A conflict occurs in ATC when two or more aircraft pass too close together. It is the air traffic controller's job to resolve potential conflicts by adjusting the trajectories of the aircraft. This decision involves: selecting the aircraft to manoeuvre, deciding on the type of manoeuvre and determining the details of the manoeuvre. The choice of aircraft to manoeuvre and the type of manoeuvre depend on several factors including:

[1] This research was carried out with the support of Eurocontrol Experimental Centre at Brétigny, a European centre for research and simulation in Air Traffic Control.

the geometry of the conflict, the capabilities of the aircraft, their position relative to their destination, etc. At the moment ISAC assists in the first two stages of this decision making process. The motivation in designing ISAC has been to reduce the decision making burden on controllers. This is of particular importance for the future as air traffic volumes increase. The current version of ISAC can resolve conflict between pairs of aircraft. However, an important design consideration has been that it should be extendible to problems involving three or more aircraft. In a CBR context this involves more complex types of case reuse. It implies that, for reasons of domain coverage, monolithic cases describing individual conflicts are not practicable.

Later in this paper we evaluate some alternatives to this monolithic case structure. In Section 2 we will describe the air traffic control problem in detail. We will also describe HIPS, a next generation visualisation aid for controllers and explain how ISAC interacts with that system. In Section 3 we will present the overall architecture of ISAC and in Section 4 we will describe the alternative case representation options that have been considered. We conclude in Section 5 with an evaluation of the effectiveness of these alternatives.

2. The problem of air traffic control

Despite the fact that modern aircraft are packed with sophisticated electronic equipment, air traffic control has always been more of an art than a science. Ground-based control essentially consists of people following the progress of aircraft represented by points (derived from radar data) on a flat display screen. The simple nature of the data available means that the controllers themselves are required to build and maintain a "mental picture" of extrapolated 4D traffic based on experience and other rather ill-defined heuristics. Having done this, the controller must mentally compare every pair of predicted trajectories to determine whether any pair of aircraft will pass within the minimum permitted separation - in which case he is required to intervene in some way to resolve the potential conflict.

Such an unscientific approach to ATC is, however, becoming less and less acceptable. Pressure for change is coming from two sources: firstly, the ATC world, as elsewhere, is undergoing an information explosion - controllers potentially have access to gigabytes of data of every sort, and the possibility to communicate with aircraft and other ground systems in ways, and at speeds, which were unimaginable when their practices were conceived. Secondly, airlines are demanding greater efficiency and quality of service from the air traffic control providers: efficiency, because ATC currently accounts for about 15% of the price of a ticket, and quality of service to allow airlines to increasingly fly their preferred (and presumably near-optimal) flight paths - this is difficult using today's practices and structures.

The problem cannot, however, be approached from a uniquely technical viewpoint. Removal of the "artisanal" aspects of ATC, particularly with regard to the task of preventing metallic contact between aircraft, touches the very heart of the profession. This means that any enhancement of the controller's skills by some type of automation must be done in a way which is sympathetic to current practices and therefore acceptable to controllers.

A number of attempts have been made to model conflict resolution activity by capturing and reproducing the rules by which controllers operate. This has inevitably failed because of the difficulty of defining clear and consistent sequences of actions. In this context it therefore seems natural to consider the use of case-based reasoning, since one of the strengths of the technique is not having to model the problem/solution domain in detail, as long as it is possible to identify the key elements involved in the process, together with an idea of their relative importance.

2.1 Principles of ATC

Commercial aircraft are controlled by ground-based air traffic controllers from the moment the engines are started at the origin of the flight to the moment the engines are stopped at the destination. To facilitate the control task once the aircraft is en-route, the airspace is divided into horizontally and vertically bounded sectors, each sector normally being the responsibility of two controllers. The size of a sector depends on the amount of traffic to be processed, the number of aircraft per hour normally being limited to around 30. This means that in areas of high traffic density the sectors will tend to be smaller, giving an average transit time of around 6 or 7 minutes, whereas in low density areas with larger sectors transit time can be around 20 minutes. Apart from national boundaries, the shape of the sector is normally a function of route structure - a sort of road system in the sky normally followed by commercial aircraft - designed so that, for example, major route crossing points do not occur near the edges of the sectors.

When an aircraft is about to enter a sector, the controller(s) responsible for that sector will be somehow notified of its arrival, and this should correspond more or less with its appearance on the synthetic radar display. A short time later they assume responsibility for the aircraft, a complementary release of responsibility having taken place in the upstream sector - the bilateral agreement of the conditions for transfer from one sector to another is known as coordination, and actually represents a substantial part of the controller’s workload.

It is then up to the controllers to see the flight through the sector, and clearly the principle concern is that the aircraft transits the sector conflict-free. There is, however, a secondary requirement which is to provide the aircraft with a cost and time-efficient passage.

2.2 Conflicts and conflict resolution

Internationally agreed rules exist defining separation standards below which aircraft are said to be “in conflict”. The values of these separations vary according to a number of factors such as the type of controlled airspace e.g. near an airport, oceanic etc. (Note that “conflict” is not synonymous with “collision”: for our purposes a conflict is rather the infringement of the applicable separation minima.) Minimum horizontal separations are typically 5 nautical miles (1 Nm=1852 m) in radar controlled regions and either 1000ft or 2000ft vertically, depending on altitude.

In practice controllers will often apply significantly larger separations, mainly due to the difficulties they have in accurately visualising future trajectories and conflict situations. This has a number of implications: for example a manoeuvre applied to resolve a conflict may end up significantly larger than is actually necessary (i.e. non-optimal), and, indeed, there will often be unnecessary intervention where, had the aircraft continued on their existing trajectories, there would not actually have been a loss of separation.

One of the most important advances in computer support for air traffic controllers in the next few years will be the provision of relatively accurate predictions of future aircraft trajectories. Such a development should, in principle, allow clearer visualisation of where aircraft will go, and in particular whether they will be in conflict. Even with such information, however, it is not immediately obvious how controllers can use it.

2.3 The HIPS System

One system which presents this information in a usable way is HIPS (Highly Interactive Problem Solver), a system developed at the Eurocontrol Experimental Centre. HIPS is a novel support tool which comprises two main parts: firstly, it displays conflict situations relative to one selected aircraft in a time-independent way, and secondly, it provides a means for the controller to modify trajectories to find solutions to conflicts. Space does not permit to describe HIPS techniques in detail, so a simple example will be used to show the principles.

Consider Figure 1. The aircraft which interests us, EEC123, is traversing our airspace from left to right. Unfortunately, its trajectory is in conflict with that of another aircraft, EEC456, which is travelling in a northerly direction. The part of our trajectory for which there is a loss of separation between ourselves and EEC456 is marked with a thicker line. If we now imagine that we want to solve this conflict by changing EEC123's heading, we could attempt various new headings assuming a certain point as our start of turn, and for each one we could check for conflicts and again mark any loss of separation in bold.

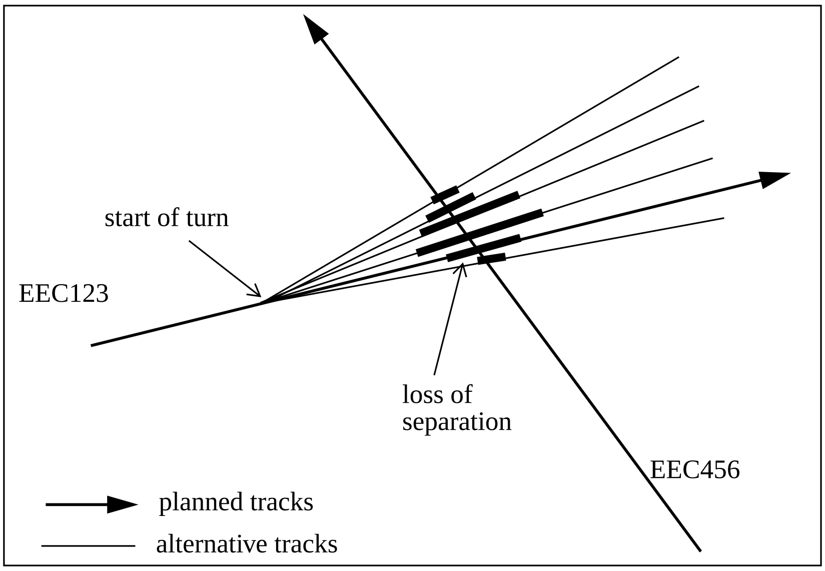

Fig. 1. A possible conflict representation**.**

Having tried a number of possibilities the next step is to group together all the bold lines to produce a single “no-go” zone as shown in Figure 2. This provides an immediate and powerful visual device by which the controller can rapidly see that in this case the conflict can be solved by a relatively small southward or larger northward deviation to EEC123.

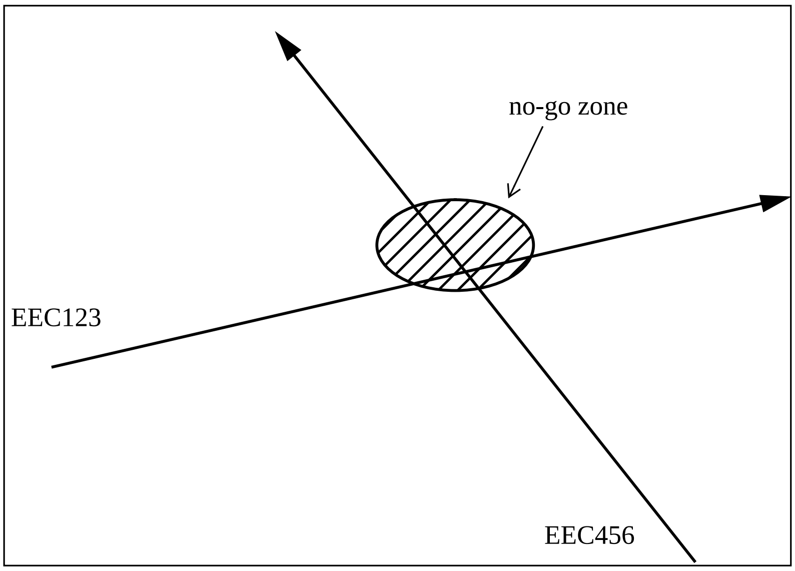

Fig. 2. How HIPS represents the conflict**.**

The above illustration assumes linear constant-speed trajectories, with the start-point of the manoeuvre already known. Unfortunately these assumptions are unrealistic in real life, which means that the techniques used for generating the diagrams are actually quite complex.

As well as generating a horizontal view, we can use a similar approach to produce diagrams for vertical and speed dimensions, giving a total of three pictures. This HIPS interface is shown in Figure 3. The Plan View Display represents the synthetic radar screen - this is the information that is available to the controller in the current generation of ATC systems. In HIPS this is augmented with the three other perspectives on the conflict. In each of these windows the conflict region is shown in dark grey (normally in red or yellow). In each case the subject trajectory can be manipulated by simple mouse interaction to determine the exact magnitude of the heading, speed or altitude change(s) required.

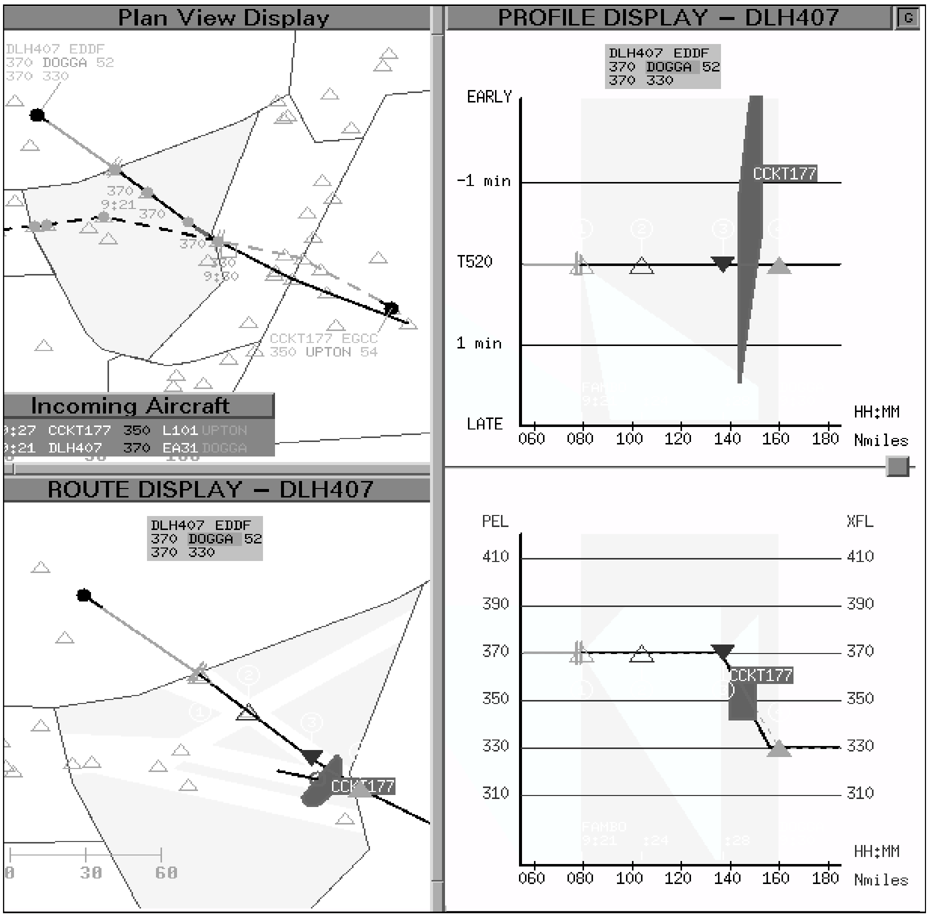

Fig. 3. The HIPS interface**.**

2.4 Conflict Resolution support for HIPS

One of the main advantages of the HIPS approach is that it does not, of itself, attempt to present complete solutions. It presents information to the controller in a way that he can (hopefully) understand, and it is then up to him to find solutions. This is a key advantage of the approach, and has been important in gaining a degree of acceptance. It has meant, however, that there are still a number of steps to be taken between the time when a potential conflict is recognised, and the implementation of the solution. In particular, the controller must:

- evaluate the conflict situation and decide which aircraft he is going to manoeuvre
- decide which type of manoeuvre is appropriate
- determine the details of the manoeuvre (e.g. turn right 10 degrees, go 10 knots faster etc.).

These decisions imply the examination of the horizontal, altitude and speed display for each aircraft involved in the conflict. The aim of ISAC (Intelligent System for Aircraft Conflict resolution) is to automatically highlight the display corresponding to the best manoeuvre of the best aircraft. This means that ISAC has to decide which is the aircraft that has to be manoeuvred and the type of manoeuvre to avoid the conflict. The solution given by ISAC can be accepted by the controller who will complete it with a deeper specification of the manoeuvre. Alternatively, it may be discarded because it is considered not adequate. If this happens, the controller will choose another display of the six available.

The main advantage in having an intelligent system behind HIPS is that the controller's workload is reduced. Moreover, the system could suggest a manoeuvre that didn't come to the controller's mind, but is more efficient.

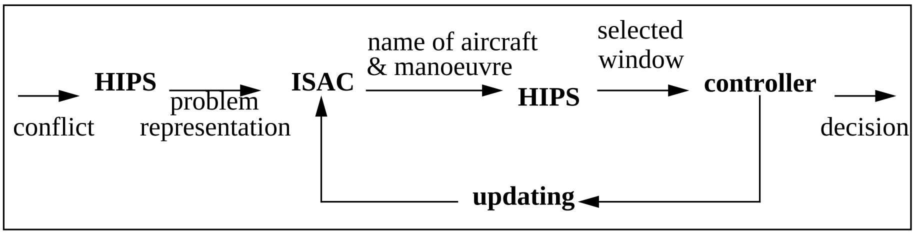

Fig. 4. How ISAC is embedded in HIPS**.**

The conflict resolution process is shown in Figure 4. When HIPS detects the conflict, it sends the description to ISAC: i.e. the flight plan and performance of the aircraft involved, the shapes of the no-go zones etc. Using this data, ISAC selects the aircraft to manoeuvre and the type of manoeuvre. Then HIPS highlights the display to be used by the controller in determining the final detail of the manoeuvre.

The controller has full visibility of all the data and has full responsibility for the manoeuvre that will be communicated to the pilot. ISAC presents only a "suggestion" of what is the best manoeuvre based on the conflict solutions stored in its knowledge base.

3. System Architecture

The current version of ISAC operates as a decision support system. It can be argued that this is the natural role for a CBR system (Kolodner, 1991). It is certainly important for its acceptance in the ATC culture that it should be a support system rather than an expert system. This means that the emphasis in ISAC is on case retrieval. However it will be seen in Section 4, there is some complexity in the reuse process in the aggregation of solutions when more than one solution is retrieved.

The retrieval process is shown in Figure 5. A key criterion in the design of the retrieval mechanism in ISAC is that it should be very fast. When a controller selects a conflict in HIPS for resolution, ISAC needs to suggest a solution very quickly. The retrieval mechanism that we have settled upon is a two stage process. These two stages reflect the fact that the case features are divided into constraints and ordinary features. The characteristics of the domain dictate that there are some features that must be matched if cases are to be considered similar. These features are considered constraints and the base filtering stage in retrieval selects cases that match on these constraints. The objective in the next stage is to select cases that match best on the remaining features. The outcome is equivalent to *k*-Nearest Neighbour (*k*-NN) retrieval but is implemented as a spreading activation process for reasons of speed.

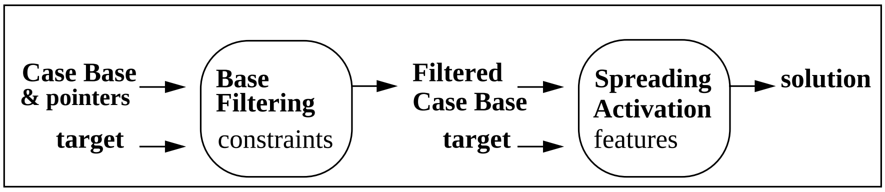

Fig. 5. The case retrieval architecture in ISAC**.**

During the HIPS start-up, ISAC loads the case base into memory and builds a web of pointers among the cases that will speed up the retrieval process. The Base Filtering mechanism discards from the case base all the cases whose constraints do not match exactly the target. This step is necessary because of the characteristics of the domain but it also has the advantage that it reduces the size of the case base before the comparatively expensive spreading activation stage. The pointers link all the cases that have the same value for a given parameter. During retrieval, activation passes along these links. The importance of the different features are weighed and activation is proportional to this importance. This implements a type of eager *k*-NN retrieval. This is appropriate because there will be a large number of retrievals in a session.

In ISAC the solution can come from one or more cases and is compound manipulable (following the convention introduced in Hanney et al., 1995). At the moment there is no adaptation because the solution required does not specify the detail of the manoeuvre and the case coverage should be sufficiently extensive that any structural transformation is not required.

4. Case Base and Case Description

Most of the development effort in ISAC has focused on the case representation. There were two problems here: the macro problem of what should constitute a case, and the micro problem of how to characterise a case.

4.1 Case Representation

It can be seen in Figure 4 that when a conflict is to be addressed a case representation is passed from HIPS to ISAC. All the data concerning the conflict are available in HIPS and are converted into parameters that describe a conflict. The conversion process eliminates useless data and transforms some others into more abstract and elaborated parameters. For example, the number of passengers on an aircraft is discarded and the coordinates of the no-go zones are meaningless per-se, but become useful if related to the aircraft trajectory. If ISAC is to propose a good solution to the conflict all relevant information must be available at this point and it must be transformed into a perspicuous representation. Determining this representation has involved extensive dialogues with ATC controllers and then the manipulation of the data from HIPS to produce the correct representation. Some examples of important parameters are:

- Horizontal Conflict Configuration: This is a general description of the conflict as one of: facing, catching, crossing, etc. It is handled as a constraint.
- Horizontal Intention: Where is the destination of the aircraft compared with its current heading?
- Near To Boundary: How close is the aircraft to exiting the sector?
- Left Exit No Go Zone: By how much does the aircraft need to turn left to avoid the no-go zone?

The process of determining these parameters is iterative with the selection of new parameters being driven by the analysis of errors at each iteration. Still it is difficult to determine a comprehensive set of important parameters from dialogues with the controllers. This difficulty is exacerbated by considerable differences in how individual controllers view conflicts. In the end there is a compromise between what is considered an important criterion and what can be extracted from the geometric information from HIPS.

4.2 Case Structure

The motivation behind the development of ISAC is to reduce the decision making burden on controllers in order to support operation in situations of increased traffic. This future scenario also implies more complex conflicts involving more than two aircraft. A key design criterion has been to develop 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. For reasons of economy in case coverage we want solutions in two-aircraft conflicts to be reusable in three-aircraft conflicts, and so on. This means that conflicts should be decomposable so that the basic unit of retrieval is an individual aircraft in a conflict.

This problem of representing cases describing two conflicting entities has already been faced in the CBR literature in two classical systems, Mediator (Simpson, 1985) and Persuader (Sycara, 1987), and more recently in Truth-Teller (Ashley, 1995). 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.

While our ultimate objective in developing ISAC is to have a single aircraft as the unit of case retrieval we have considered three case organisations in detail. We have evaluated two alternatives with two aircraft per case and one alternative with one aircraft per case. The option of storing the two conflicting aircraft in the same case (TwoInOne) is the most obvious because it reflects the controller's way of examining a conflict, but it presents the problem of deciding which is the first aircraft in the conflict description. Let us suppose that a conflict between two aircraft A1 and A2 is stored in the case-base in the form A1A2. If the same conflict is to be solved again, HIPS will give ISAC the description of A1 and A2, but ISAC could build either target A1A2 or target A2A1, inverting the order. If this happens the probability of finding the correct case A1A2 in the case base is very low. This problem is called the "symmetric problem". An obvious, but time and space consuming, solution to the symmetric problem is to build the two cases A1A2 and A2A1 for each conflict in the case base, that means a case base that is twice the size it needs to be. This means that, in our implementation, retrieval will take twice as long. Alternatively, we may build two targets P1P2 and P2P1 for each conflict between P1 and P2 and repeat the retrieval process twice, that means a doubled retrieval time. The advantage of both these solutions is that there is no knowledge loss. We actually evaluated the first option with the bloated case-base.

An alternative solution is to produce a set of rules that will decide which is the first and the second aircraft in the case description. These rules will have to be applied during the case-base construction and every time a new target problem is presented. After the rules are applied, the case is said expressed in the "canonical form". The advantage of this process is that neither the retrieval time nor the case base dimension is doubled. This involves the risk of loss of information as we can see from the experiments results.

The final means of representing a conflict is by storing the information on each aircraft, plus a general description of the conflict, in a separate case (OneInOne). The information of the other aircraft involved in the conflict is implicit in the environment description in the form of the no-go zones (see Table 1). This moves us towards a Hierarchical CBR situation (Smyth & Cunningham, 1992) 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. Moreover, the "symmetric problem" is avoided. However we found it more difficult to come up with a set of parameters that capture all the details.

Evidently this last approach requires that a conflict be represented as two (or more) target cases and that there should be a retrieval for each one. This requires a policy for the extraction of the final solution from the two sets of matching cases, because the two solutions could lead to an incongruence. In fact, this is a delicate issue. Let us suppose that in the case base there are two conflicts A-B and C-D which are represented in four cases A, B, C, D. If a new conflict is X-Y, ISAC will build two cases X and Y and will start the retrieval process. If the conflict X-Y is, in general, more similar to the conflict A-B, it does not imply that the retrieved cases will be A and B, because the retrieval results depend on the individual aircraft matching. Our current policy is to select the highest scoring case but more experimentation is required on this issue.

Table 1. Example of an aircraft description in the OneInOne case representation**.**

CaseName	CCKT177_2764
HorConflConf	crossing
SameDestination	no
AltitudeNow	lower
AltIntention	StableDescend
HorizIntention	LeftLeft
Speed	faster
DimAltZone	big
Performance	worse
CloseToBoundaries	yes
MilesDone	1741
MilesToDo	147
RightExitNoGo	53.03
RightAvailable	36.97
LeftExitNoGo	26.15
LeftAvailable	63.85
ReqLevFree	Above&Below
Faster	0.53
Slower	1.58
InFrontDirect	yes
InFrontMoreSpace	yes
@s hor3*alt2	

It can be seen that all the values are symbolic (enumerated). This is the result of the iterative research done with the controller. Initially the case representation contained a lot of numeric values but it became evident that the controller considered values in ranges so it was decided that symbolic values were more suitable.

5. Evaluation

ISAC is currently in the second phase of its evaluation. In the first phase it was evaluated on what might be called iconic conflicts. These were artificially constructed conflicts and the case-base captured all the basic ways in which two aircraft can approach too close together. There are two components to a solution in ISAC. It must nominate the aircraft to manoeuvre (A, B or A&B) and it must select the type of manoeuvre (horizontal, altitude or speed). After some refinement of the case representation ISAC performed very well in this initial evaluation. With a case-base of 50 conflicts and a set of 10 parameters, using a leave-one-out evaluation strategy, all three case organisations described in section 4.1 were able to produce good solutions in greater than 95% of the cases.

It is clear that these artificially constructed cases do not capture the full complexity of real ATC conflicts. When this case organisation was evaluated on conflicts taken from real traffic samples the number of completely acceptable solutions dropped to little over 51% and either the correct aircraft or the correct manoeuvre was selected in little over 70% of cases (i.e. 70% were half right). In resolving conflicts in real situations with complex trajectories there are details that were not represented in this initial case representation. With a refining of the acquisition of the parameters and with the acquisition of a new case base, coming from more realistic conflicts, these two numbers reached 75% and 85% with the OneInOne case organisation.

In the Figures 6 and 7, we report the results obtained from a case base of 51 conflicts described with 18 basic parameters (some of these parameters are duplicated in the TwoInOne case representation). It should be noted that the evaluation and refinement process is still ongoing. We are confident that we will be able to produce solutions that will be accepted by the controller in greater than 95% of situations.

A lot of work still has to be done on extending the case base, because 50 conflicts are not enough to characterise all the possible ATC conflicts, and on the parameters acquisition, because it is not always obvious how to convert into numbers what the controller sees on the radar screen.

In addition to improving our case representation in our evaluation process we are also interested in discovering which parameters are particularly predictive and which should be handled as constraints. In Figure 6 we show that handling one of the parameters as a constraint improves the competence of the system slightly. This is in addition to improving the retrieval time.

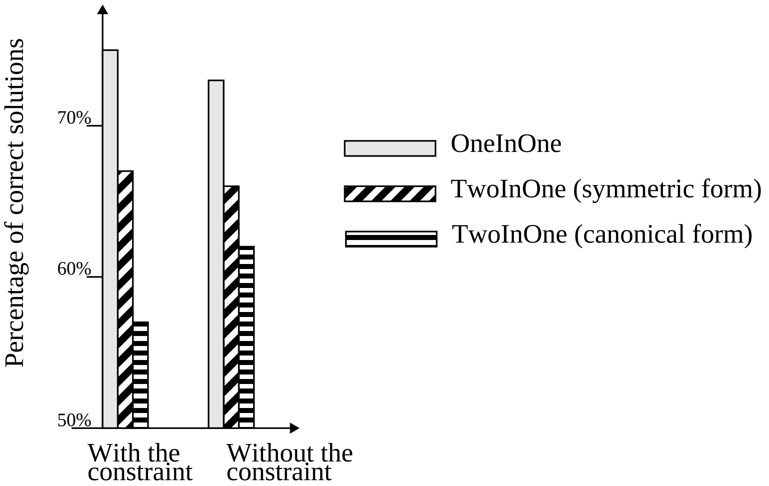

Fig. 6. Advantages in using the constraints**.**

We are also interested in simplifying the case representation so that the case description would involve less parameters. To this end we have evaluated the discrimination power of all the parameters using the mechanism used in ID3 (Quinlan, 1986) and have evaluated retrieval while ignoring less discriminating parameters. In Figure 7 we can see that removing one parameter actually improves performance but after that, performance begins to deteriorate.

From both Figures 6 and 7 it can be seen that the performance of the canonical form is worse than the performance of the symmetric form because of the loss of information. Nevertheless, we still think that the option of using the canonical form is not to be discarded because, with a more effective policy for ordering the aircraft, the results may improve.

Conclusions and Future Work

The performances of CBR systems in general and of ISAC in particular depend on how well the case is described and on how densely and homogeneously the case space, i.e. the set of all the possible conflicts, is populated.

To populate our case space it was first necessary to list what were the most common and realistic conflicts - these we called “gold standard cases”. These cases should cover all the case space. Using the terminology of (Smyth, 1995) these cases should be pivotal, i.e. if a pivotal case is deleted the competence of the system is reduced, but the regions of these pivots should cover all the case space.

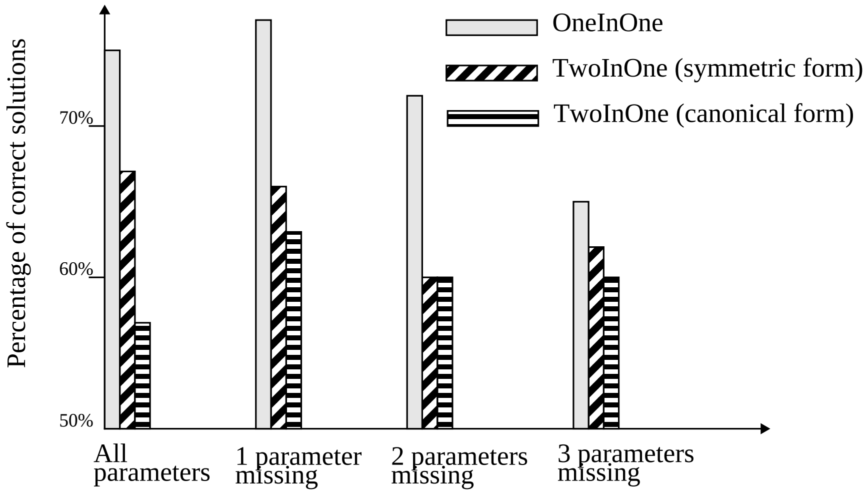

Fig. 7. Performance compared to the number of parameters**.**

The idea to create from scratch a traffic sample containing all the gold standard conflicts was rejected after some tests: the controllers judged the environment too unrealistic and thus the solutions given. It was necessary to introduce a more realistic display environment and the use of more elaborated traffic samples (coming from real-time simulation) to finally bring to the controller an environment that could enable him to give “reliable” solutions. We are now close to settling on a definitive case representation. Once this final case representation has been determined, our long term objective is to extend the system to tackle conflicts with more than two aircraft.

Acknowledgments

We would like to thank Nigel Makins and Andrew Barff for sharing their expertise in Air Traffic Control with us.

References

Ashley K.D., McLaren B.M., (1995), Reasoning with Reasons in Case-Based Comparisons, Case-Based Reasoning Research and Development, Lecture Notes in Artificial Intelligence, M. Veloso & A. Aamodt (eds), pp133-144, Springer Verlag.

Hanney K., Keane M., Smyth B., Cunningham P., (1995), Systems, tasks and adaptation knowledge: Revealing some revealing dependencies, Case-Based Reasoning Research and Development, Lecture Notes in Artificial Intelligence, M. Veloso & A. Aamodt (eds), pp461-470, Springer Verlag.

Kolodner J.L., (1991), Improving Human Decision Making Through Case-Based Decision Aiding, AI Magazine, Vol. 12, No. 2, Summer 1991, pp52-68.

Meckiff C. and Gibbs P., (1994), PHARE Highly Interactive Problem Solver, EEC Report 273/94 [on request to Colin.Meckiff@eurocontrol.fr].

Quinlan J.R., (1986), Induction of Decision Trees, Machine Learning, 1, 81-106.

Simpson R.L., (1985), A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute mediation, PhD thesis, Georgia Institute of Technology, TR. GIT-ICS-85/18.

Smyth B., Cunningham P., (1992), Déjà Vu: A Hierarchical Case-Based Reasoning System for Software Design, in Proceedings of 10th. European Conference on Artificial Intelligence, Vienna, Austria, ed. Bernd Neumann, Wiley & Son, pp587-589.

Smyth B. and Keane M.T., (1995), Remembering to forget: A competence-preserving deletion policy for CBR systems, C Mellish (Ed.), Fourteenth International Joint Conference on Artificial Intelligence. Los Altos: Morgan Kaufmann.

Sycara E.P., (1987), Resolving Adversarial Conflicts: An Approach to Integrating Case-Based and Analytic Methods, PhD thesis, Georgia Institute of Technology, TR. GIT-ICS-87/26.