

Knowledge Engineering Requirements in Derivational Analogy

Pádraig Cunningham*, Donal Finn**, Seán Slattery*

*Department of Computer Science, Trinity College Dublin, Ireland

**Hitachi Dublin Laboratory, Trinity College Dublin, Ireland

Abstract. A major advantage in using a case-based approach to developing knowledge-based systems is that it can be applied to problems where a strong domain theory may be difficult to determine. However the development of case-based reasoning (CBR) systems that set out to support a sophisticated case adaptation process does require a strong domain model. The Derivational Analogy (DA) approach to CBR is a case in point. In DA the case representation contains a trace of the reasoning process involved in producing the solution for that case. In the adaptation process this reasoning trace is reinstated in the context of the new target case; this requires a strong domain model and the encoding of problem solving knowledge. In this paper we analyse this issue using as an example a CBR system called CoBRA that assists with the modelling tasks in numerical simulation.

1 Introduction

Case-Based Reasoning (CBR) has emerged from research in cognitive psychology as a model of human memory and reminding. It has been embraced by researchers on 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. One of the central advantages in using a case-based approach to developing knowledge-based systems (KBS) is that CBR systems can be developed without encoding a strong domain theory for the problem domain [1]. In particular CBR systems can be developed without explicit encoding of problem solving knowledge. However there are CBR systems that incorporate a strong domain theory. Systems that set out to support a sophisticated case adaptation process do require a strong domain model. So there is some question as to whether these CBR systems with deep knowledge representations lose this central advantage of the CBR approach to KBS development.*

The Derivational Analogy (DA) approach to CBR is a case in point [2-7]. In DA the case representation contains a trace of the reasoning process involved in producing the solution for that case. In the adaptation process this reasoning trace is reinstated in the context of the new target case. If the domain model is to support the reinstatement of a reasoning trace then it will have to be a fairly comprehensive representation.

In this paper we will attempt to analyse this issue using as an example a CBR system called CoBRA (Case-Based Reasoning Assistant) that assists with the mathematical modelling tasks in numerical simulation. CoBRA is a DA based CBR

* It is worth mentioning that these comments apply to the use of CBR in developing KBS and not to the use of CBR as a model of memory and mental processes.

system that produces simplified models of cooling fins for heat and fluid flow analysis. CoBRA's cases consist of model descriptions and a trace of the model simplification process (see Figure 4). The adaptation process attempts to reapply this reasoning trace to the fin model in the target case. This regenerative adaptation involves the reworking of problem solving knowledge so it is necessary that domain knowledge and problem solving knowledge be encoded in the system.

The questions we wish to consider are:-

- (i) Is the advantage of CBR lost in having to support it with a deep model of the problem solving process?
- (ii) Does the use of CBR in complex problem domains manage to avoid any of the knowledge engineering needed for a solution based on reasoning from first principles?
- (iii) Could this system have been developed as readily by encoding the knowledge as a planning system of transformation operators?

As a counterexample to this complex CBR application we will describe Rachmann, a system for property valuation. As an example of a CBR system using substitution adaptation this is at the easy end of the spectrum and the advantage of CBR is evident. We will argue that substitution adaptation works for the valuation task because the problem structure is simple; an important characteristic being that the solution is atomic. In tasks where the solution expression has a complex structure the adaptation process is more delicate and interaction between solution components must be considered. The cases in CoBRA have this kind of complexity.

In the next section we analyse the characteristics of the different types of CBR systems, focusing on the knowledge acquisition requirements of the different adaptation strategies. In section 3 substitution adaptation and transformation adaptation are discussed in this perspective. The adaptation process in CoBRA is described in section 4. In section 5 we discuss the knowledge engineering requirements to support generative adaptation.

2 CBR and Problem Complexity

The basic tenet of CBR is that, rather than solve a problem from first principles, it may be easier to retrieve a similar problem and transform the solution to that problem. In Figure 1 we attempt to illustrate this trade-off graphically. SP' represents the specification for a new problem and SL' is the solution to that problem. FP' represents the search process that establishes this solution from first principles—the task we wish to avoid. A CBR solution is worthwhile if the retrieval task R, and the adaptation task A are simpler than FP'.

From an 'AI as Engineering' perspective the big issue here is the complexity of the adaptation task A. An analysis of the CBR literature suggests that CBR adaptation might be divided into three categories arranged in order of increasing complexity as follows:-

- **Substitution Adaptation:** This is the simplest type of adaptation and merely involves substituting some of the parameters in the solution. The Rachmann system described in section 3.1 is a typical example of a CBR system requiring this minimal adaptation.

- **Transformational Adaptation:** This adaptation is more complex and involves structural changes to the solution. (see [8], [1])
- **Generative Adaptation:** This is the most complex adaptation and is represented by A^G rather than A in the diagram. The adaptation process involves a reworking of the reasoning process FP in the context of the new problem situation represented by SP' . Generative Adaptation is also known as Derivational Analogy. [2-7]

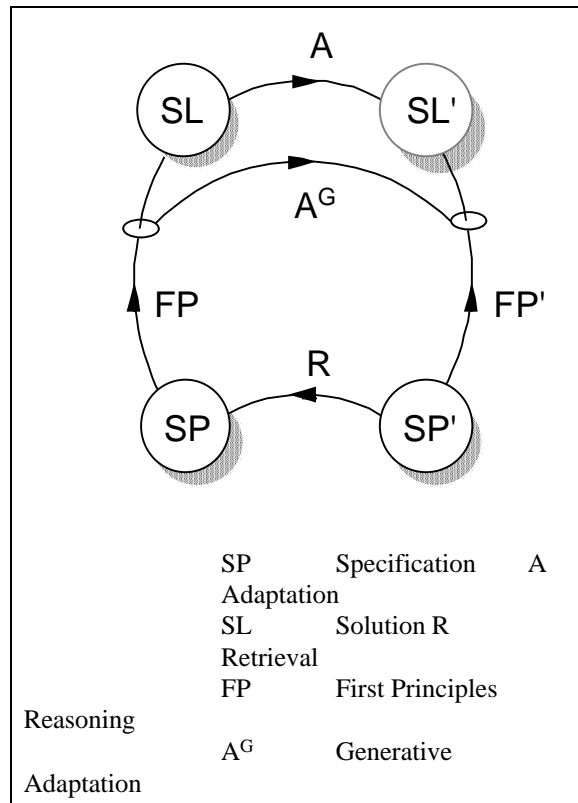


Figure 1. The transformation processes in CBR and in reasoning from first principles.

These different adaptation categories are appropriate for problems of different complexity. Substitution adaptation will only work for comparatively simple problems where the solution statement is simple or atomic (expressible as a single price or a fault category for instance). Transformation adaptation can work where the solution has a more complex structure (a plan perhaps) but the components of the solution are not very interdependent. Thus the distinction between substitution and transformation is one of *degree*. The other aspect to this is that transformation offers more coverage than substitution. Cases can be transformed to a wider variety of different solutions but a more complete domain model is required to do so (see Figure 2).

For problems where the solutions are made up of interdependent components, as occurs in design for instance, solutions are too *brittle* to be transformed in this

manner [3]. Instead, it is necessary to re-generate solutions as is done in Derivational Analogy.

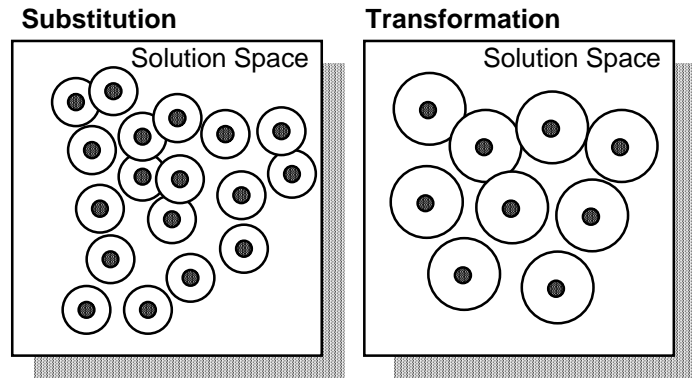


Figure 2. Transformation adaptation has more coverage than substitution. With transformation cases can be adapted to a wider variety of solutions.

This diversity in CBR means that the different categories of CBR systems have radically different characteristics. Some of these differences are summarised in Table 1. In CBR systems supporting substitution adaptation, retrieval is based on surface features and only a minimal amount of domain knowledge is encoded in the system. Transformation adaptation requires retrieval based on more abstract features and needs access to a deep domain model. Derivational Analogy involves the replay of a reasoning trace so problem solving knowledge is encoded in the case base. Since it is accepted that problem solving knowledge is considerably more difficult to acquire than domain knowledge (see arguments of Bergmann et al. [9]) this means that DA is radically different from other approaches to CBR from the perspective of knowledge acquisition. This is consistent with Veloso's view that DA is a bridge between memory based reasoning (in the broad sense) and traditional first principles planning [6].

Table 1. A summary of the different categories of CBR system.

Adaptation	Retrieval	Knowledge Representation
Substitution	Based on Surface Features	Minimal Domain Knowledge
Transformation	Based on more Abstract Features & Goals	Significant Domain Knowledge
(re)Generation	Based on Problem Features, Goals & Justifications	Domain Knowledge & Problem Solving Knowledge

3 Substitution & Transformation Adaptation

The basic argument emerging from the analysis above is that there is a fundamental difference between CBR systems involving substitution or transformation adaptation and systems involving regenerative adaptation. The fact that the later involves encoding problem solving knowledge results in knowledge acquisition requirements that are considerably more complex. Before moving on to discuss Derivational Analogy in CoBRA we will describe how a CBR system for property valuation can operate even though there is no explicit encoding of problem solving knowledge.

3.1 Rachmann: A classic CBR system

Rachmann is a small CBR system for property valuation. Each case is a property represented as set of features and the value of that property (see Figure 3). A target case is a set of features representing a property for which a valuation is sought. The system finds the best match from its case base and performs simple adaptations on that case to determine a valuation for the target case.

The advantages of CBR for knowledge acquisition are manifest in this example. The cases are easy to set up as the features are obvious important attributes of houses affecting the market value. However, the system is not completely without a domain theory because the organisation of the indices in the discrimination network reflects their relative importance. In addition, the partitioning of a city into locations reflecting property values requires some expertise. So the domain model in this system is a frame representation describing houses locations and important features. For this task problem solving knowledge might be encoded as heuristics that link predictive features to property values. However, in this system there is no explicit encoding of problem solving knowledge; instead the case retrieval mechanism has implicitly learned these relationships. For this reason the system has been much easier to set up than an equivalent system using heuristics to reason from first principles.

4WF	<i>Indices</i>	3 LR	<i>Indices</i>
	Location: SM-1		Location: SM-1
	B-Rooms: 2		B-Rooms: 3
	Age: Modern		Age: Modern
	Rec-Rooms: 1		Rec-Rooms: 2
	Kitchen: Small		Kitchen: Large
	Rear-Acc.: No		Rear-Acc.: Yes
	Tot-Area: <800		Tot-Area: >1,200
	En-Suite: No		En-Suite: Yes
	: : : :		: : : :
	Price £75,000		Price £98,000

Figure 3. Example cases from Rachmann, the property valuation system

This ability of CBR systems to operate without problem solving knowledge continues to hold true for transformation adaptation as long as the components of the solution are not very interdependent (see CHEF [8] or Déjà Vu [1]). However, as we shall see in the next section DA does not have this simplicity.

4 Generative Adaptation in CoBRA

CoBRA is an example of a CBR system operating in a domain where substitution adaptation is inadequate. The application domain of CoBRA is mathematical modelling in numerical heat transfer analysis which is considerably more complex than the Rachmann house valuation domain. Heat transfer problems are described mathematically by the thermal partial differential equations (PDEs) and are usually analysed using numerical simulation techniques such as the finite element method. Modelling precedes numerical analysis and involves abstracting a mathematical model from a real world problem by applying physical and mathematical idealisations. The objective for modelling is to create a model that is computationally realistic to solve, but, at the same time retains the important features of the physical system. This task forms an important initial stage in real world engineering analysis problems and for the analyst it involves making a series of assumptions and justifications to produce the simplified model [10,11,12].

Figure 4 illustrates a typical modelling scenario associated with the analysis of an electronic cooling fin. This problem is considered too complex to analyse directly and must therefore be modelled before numerical analysis can be undertaken. Cooling fins can be modelled by applying one of the following strategies; fin simplification, fin removal and compensation (with an equivalent boundary condition) or complete fin removal.

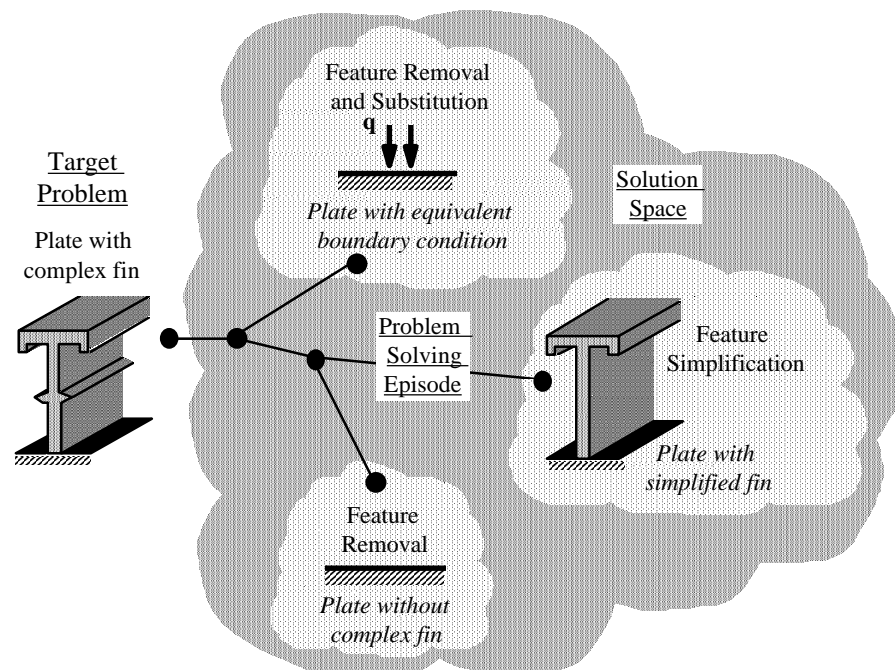


Figure 4 Model Reasoning Trace in CoBRA

The system has been implemented in frames in Common Lisp and uses an AutoCAD interface to communicate model changes to the user. Execution involves

selecting suitable cases and reworking the derivational trace associated with that case in the new context. The system narrates its rework as follows:-

Checking predicates:

- Feature must face into flow
- Feature must be centred on face
- Feature must be compared to the width of the face on which it sits
- Feature height must equal its width

All predicates checked: applying operations.

*** Feature not significant from flow perspective ***

Checking predicates:

- The feature's exposed surface area must be small compared to the surface area of the face on which it sits.

All predicates checked: applying operations.

*** Feature not significant from heat perspective ***

*** Removing ***

Modelling strategies are based on experiential modelling episodes and therefore do not have a strong domain theory [11]. The reasoning process involved in deciding which strategy to apply requires the use of fundamental heat transfer knowledge in the form of formulae, approximations, correlations and assumptions. This knowledge is problem solving knowledge, and it provides the means to assess the applicability of a strategy in the context of the problem under consideration. Although this problem solving knowledge has a strong theoretical background and can be considered to be relatively complete, its application is poorly understood and successful use is based on experience. Furthermore, the application of such problem solving knowledge is highly dependent on the physical system being analysed and the strategy being assessed. The implication of this is that the use of model based or generalised planning techniques have proven to be problematic in model generation [11]. This has led us to examine the use of derivational analogy techniques in this domain which has been tackled using planning and first principles approaches hitherto.

4.1 Case Descriptions in CoBRA

The nature of the reasoning process associated with problem solving in CoBRA required the use of generative adaptation techniques, which involved the encoding of problem solving knowledge as well as domain knowledge within the cases. In CoBRA, a case consists of a representation of the real world physical system, the solution in the form of a simplified model and a reasoning trace of the justifications for the transformations in going from the real world problem to the simplified model. Figure 5 illustrates a portion of such a case. The diagram on the left shows a cross section of a finned heat exchanger cooler, and the task addressed by CoBRA is to produce a simplified model of this physical system. The frame definition on the right illustrates the problem description, the problem solution and the derivational trace that provided this solution. A target case contains only the problem description; this is the specification of the physical system. Cases are retrieved using an activation network based on feature similarities and a case solution is created by using generative adaptation involving a re-run of the reasoning trace. This adaptation by regeneration is derivational analogy.

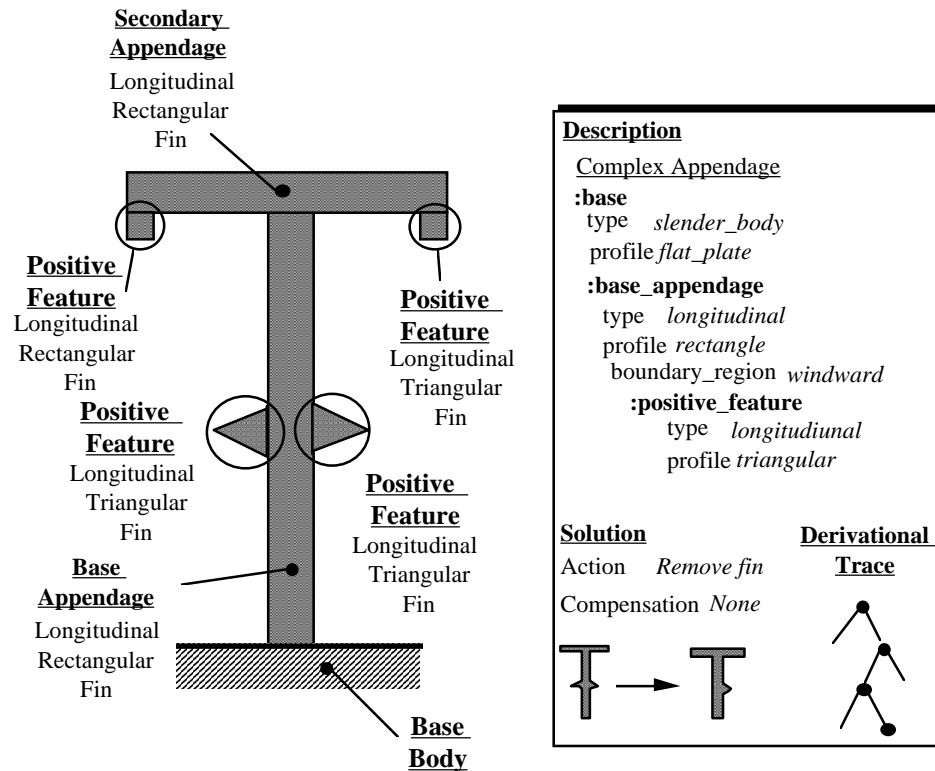


Figure 5 A case in the CoBRA system

4.2 Regenerative CBR in CoBRA

In CoBRA, the derivational trace links the start and goal state of a case. Each reasoning trace has two main components; a *decision part* and a resulting *action part* (after [2]). The decision part contains:

- Alternative modelling strategies considered and rejected
- Assumptions and justifications for the decisions taken.
- Heat transfer domain knowledge describing dependencies of later decisions on earlier ones.

The action part holds the steps taken as a result of the reasoning trace of the decision part. A typical action is, "Remove the feature which faces into the flow". The main actions in CoBRA are REMOVE, RESIZE and COMPENSATE. The actual functions used to express these actions must be sufficiently abstract to allow their application to cases similar to the one with which they are stored. Both the decision and action parts operate on parameters which are common to all cases, for example:

- **surface-area:** the heat transfer surface area
- **base-area:** the surface area of the feature base.

- **location:** the location of a feature

A typical reasoning trace is shown in Figure 6. Each node in the reasoning trace represents a decision point in the model simplification process. Goal_1 and Goal_2 illustrate how a reasoning trace in derivational analogy represents a known good route through a vast search space. Goal_3 shows the various fin modelling strategies that are considered and the actions associated with each strategy. In this situation the modelling strategies depend on the amount of heat transfer associated with the feature under consideration. By estimating this heat loss parameter a suitable strategy can be chosen and the appropriate modelling actions can be applied to the target case.

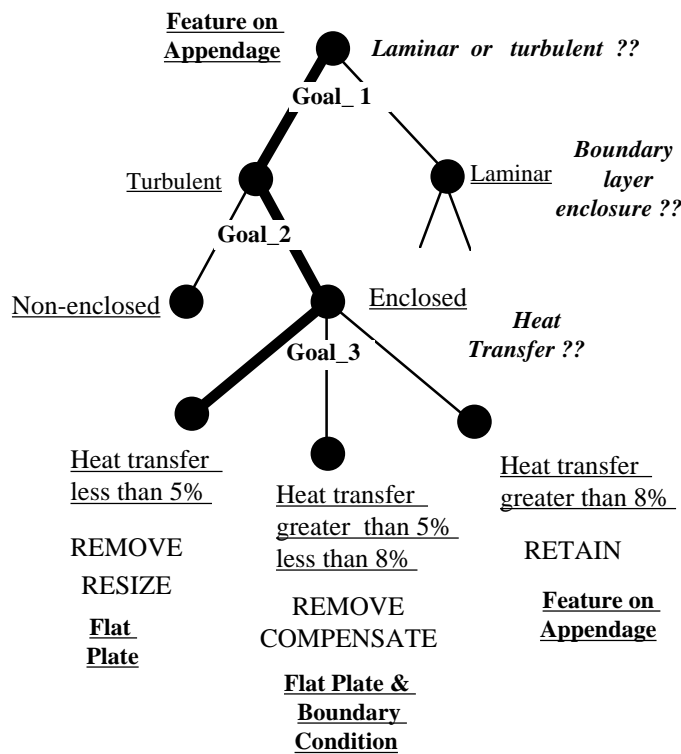


Fig. 6 A derivational trace for a windward finned appendage

5 Discussion

In the introduction, we posed three questions on the use of generative techniques in CBR applications. These focused on the use of generative adaptation in CBR, the knowledge acquisition process in generative adaptation and the development of CBR systems that are based on generative adaptation. We deal with these issues in the following sections.

5.1 Generative Adaptation in CBR

The question as to whether generative adaptation compromises some of the advantages associated with CBR arises because of the need for a deep model to support this category of adaptation. This is because CBR has the important advantage that case-bases are considerably easier to set up than other knowledge representations. There are other advantages however. In problem solving, cases encode known good routes in the solution space thereby reducing backtracking. These advantages will only be maximised in CBR systems where solution representations are not made up of complex interacting components and the adaptation process is comparatively simple. Our experience with the Rachmann system confirms this view. However, in CoBRA, where solutions have complex representations, adaptation is more difficult and a full domain model is required to support this adaptation, our opinion is that some of these advantages have been lost.

5.2 Knowledge Acquisition in Generative Adaptation

We would like to consider two perspectives on the issue of knowledge acquisition. Firstly, from the CBR viewpoint, we compare the knowledge acquisition process in systems based on either substitution or generative adaptation. Secondly, complex domains (like CoBRA) can also be implemented using model based or planning approaches; we examine from a knowledge engineering perspective the knowledge acquisition task in generative CBR and model based approaches.

Considering the CBR perspective, in the Rachmann case-base, setting up the system required very little knowledge level analysis as the heuristics were encoded *implicitly* in the cases. By contrast a rule-based system for the same task would require the determination of the influence of features such as; location, facilities, etc. would have on price. So, for this property valuation problem, CBR has avoided this need to explicitly encode a domain model. In CoBRA however, the knowledge engineering process required the *explicit* acquisition of problem solving knowledge from a domain expert and representation of this knowledge in the reasoning traces.

From a knowledge engineering perspective, the conventional alternative to the CoBRA system involves developing a model of the entities in the problem domain and encoding heuristics that represent the transformations on these entities. The granularity of the knowledge required for a model-based approach is similar to that encoded in the reasoning traces in CoBRA. However, it can be said in favour of generative CBR that the emphasis on cases focuses the knowledge acquisition process and as a result provided no special difficulties for our domain expert. This is in contrast to experiences for elicitation of generalised knowledge in model based reasoning systems.

5.3 System Issues

Given that for complex applications, the granularity of knowledge representation in generative CBR systems is comparable to that in model based approaches, the issue arises as to whether there are any control advantages associated with derivational CBR. It is evident from the literature that there has been much research on complex application domains that do *not* use CBR, particularly under the headings of model

based reasoning and qualitative reasoning (see for instance [13-15]). This research emphasises representation but it is evident that the reasoning process in mind is one of planning, with search involving backtracking through a solution space. Our experience is that the reasoning traces in DA can help constrain this search process by providing known good routes through complex solution spaces.

6. Conclusions

In this paper we analysed the issue of generative adaptation in CBR systems that use derivational analogy. This study was based on comparing CoBRA, a generative CBR system with the Rachmann system which is based on substitutional adaptation. Our findings are that for complex application domains, using a CBR approach still requires the *explicit* acquisition of problem solving knowledge which must be encoded as derivational traces. In our opinion, this overhead compromises the important advantage associated with substitutional CBR systems where domain knowledge is encoded implicitly in the cases. On the other hand, for complex application domains, generative CBR may offer some advantages compared to model based reasoning approaches. Our experience has been that episode-based problem solving knowledge proved easier to acquire than generalised problem solving knowledge, and secondly, derivational traces avoid the extensive search and backtracking associated with planning based systems by encoding a known good route through a vast solution space.

References

1. Smyth B., Cunningham P., "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, 1992.
2. Carbonell J.G., "Derivational Analogy: A theory of reconstructive problem solving and expertise acquisition", in *Machine Learning Vol. II*, R.S. Michalski, J.G. Carbonell, T.M. Mitchell eds., pp371-392, 1986.
3. Mostow J., "Design by Derivational Analogy: Issues in the automated replay of design plans", *Artificial Intelligence*, pp119-184, Vol. 40, 1989.
4. Bhansali S., Harandi M.T., "Synthesis of UNIX programs using derivational analogy", *Machine Learning*, pp7-55, Vol. 10, 1993
5. Carbonell J.G., Veloso M., "Integrating derivational analogy into a general problem solving architecture", *DARPA Workshop on Case-Based Reasoning*, pp104-124, Clearwater Beach, Florida, Morgan Kaufmann, 1988.
6. Veloso M., Carbonell J.G., "Learning analogies by analogy - the closed loop of memory organisation and problem solving", *DARPA Workshop on Case-Based Reasoning*, pp153-158, Clearwater Beach, Florida, Morgan Kaufmann, 1989.

7. Veloso M., Carbonell J.G., "Learning by analogical replay in PRODIGY: first results", *European Working Session on Learning*, Y. Kodratoff, ed., pp375-389, Porto, Portugal, Springer Verlag, 1991.
8. Hammond K.J., *Case-Based Planning: Viewing Planning as a Memory Task*, Academic Press, New York, 1989.
9. Bergmann R., Pews G., "Explanation-based similarity for case retrieval and adaptation and its application to diagnosis and planning tasks", in *Working papers of European Workshop on CBR*, Universität Kaiserslautern SEKI Report SR-93-23, pp301-306, 1993.
10. Finn D.P., Grimson, J.B., and Harty, N.M. "An intelligent modelling assistant for preliminary analysis in design", In: J.S. Gero (ed) "Artificial Intelligence in Design" Proceedings of the 2nd Int. Conf. of Artificial Intelligence in Design, pp 579, 596, (1992).
11. Finn, D.P. "A Physical modelling assistant for the Preliminary Stages of Finite element Analysis" *Artificial Intelligence in Engineering Design, Analysis and Manufacturing (AI EDAM)*, 7(4), 1993.
12. D. Finn, S. Slattery, P. Cunningham, "Modelling of Engineering Thermal Problems - An Implementation using CBR with Derivational Analogy" in *Proceedings of European Workshop on Case-Based Reasoning*, Kaiserslautern, Germany, November, 1993.
13. K. Yip: "Model simplification by asymptotic order of magnitude reasoning" In: D. Weld (ed.): *Working Papers Qualitative Reasoning '93 (QR '93)*. Seattle: University of Washington 1993, pp. 266-272.
14. S. Ling, L. Steinberg, Y. Jaluria: MSG: "A computer system for automated modelling of heat transfer" *Artificial Intelligence in Engineering Design, Analysis and Manufacturing (AI EDAM)*, 7(4), 1993.
15. S. Addanki, R. Cremonini, J. Scott Penberthy: "Reasoning about assumptions in graphs of models". In: D. Weld and J. de Kleer (eds.): *Qualitative Reasoning about Physical Systems. San Mateo: Morgan Kaufmann 1990*, pp. 546-552.