In: B. Bartsch-Spörl, D. Janetzko & S. Wess (eds.), Proc. of the 3rd German Workshop on CBR, March 1995, LSA-Report 95-02, Centre for Learning Systems and Applications, University of Kaiserslautern, pp. 63-72

Case-Based Reasoning for Decision Support and Diagnostic Problem Solving: The INRECA Approach

Klaus-Dieter ALTHOFF[1], Eric AURIOL[2], Ralph BERGMANN[1], Sean BREEN[3], Stefan DITTRICH,[4] Roy JOHNSTON[3], Michel MANAGO[2], Ralph TRAPHÖNER[4], Stefan WESS[1]

1 Introduction and Overview

INRECA offers tools and methods for developing, validating, and maintaining decision support systems. INRECA's basic technologies are inductive and case-based reasoning, namely KATE-INDUCTION (cf., e.g., Manago, 1989; Manago, 1990) and S^3-CASE, a software product based on PATDEX (cf., e.g., Wess, 1991; Richter & Wess, 1991; Althoff & Wess, 1991). Induction extracts decision knowledge from case databases. It brings to light patterns among cases and helps monitoring trends over time. Case-based reasoning relates the engineer's current problem to past experiences. INRECA fully integrates both techniques within one environment and uses the respective advantages of both technologies. INRECA offers hypermedia interfaces and graphic browsers. Its object-oriented representation language CASUEL (cf. Manago, Bergmann et al., 1994) allows the definition of complex case structures and relations. The domain model editor is used to interactively define classes, objects, attributes, and relations. The case manager uses this information to build automatically a questionnaire for collecting cases (cf. Fig. 1). The questionnaire can be customised and hypermedia entities - such as sound, drawings, pictures, and videos - may be integrated.

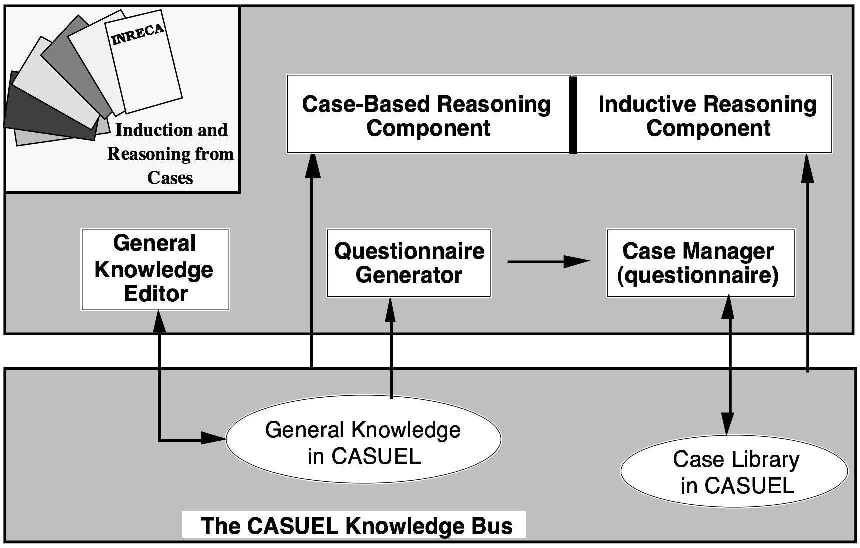

Fig. 1. INRECA Overview

In the following sections, we will give a rough overview of INRECA. After a short description of its system architecture we focus on its representation capabilities for case-specific as well as general knowledge. This is completed with an example for using CASUEL and a short overview of similarity assessment in INRECA. In section 7, we describe the integration of induction and case-based reasoning within INRECA in some detail. Sections on application domains and case-based reasoning tool evaluation support the discussion of the achieved results. Finally, we give a short summary and some basic conclusions.

[1] University of Kaiserslautern, Dept. of Computer Science, PO Box 3049, 67653 Kaiserslautern - Germany.

[2] AcknoSoft, 58a rue du Dessous des Berges, 75013 Paris - France.

[3] Interactive Multimedia Systems, Clara House, Gleanageary Park, Gleanageary, Co. Dublin - Ireland

[4] tecInno GmbH, Sauerwiesen 2, 67661 Kaiserslautern - Germany.

2 System Architecture

INRECA offers its user the option to consult the system in different manners. The underlying data structure is *one* tree, called *k*-d tree (cf. Wess, 1995). But, this tree is able to behave like a decision tree as well as an indexing tree to support case-based reasoning. Within the tree, symbolic and numeric attributes can be represented and efficiently handled as well as unknown values and unordered value sets. In addition, INRECA offers sequential retrieval and value-based indexing for relational retrieval. Fig. 2 and 3 show the overall use of the consultation system and the system builder.

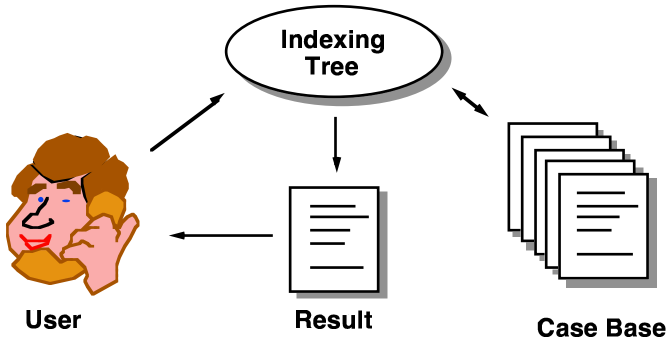

Fig. 2. Consultation with INRECA

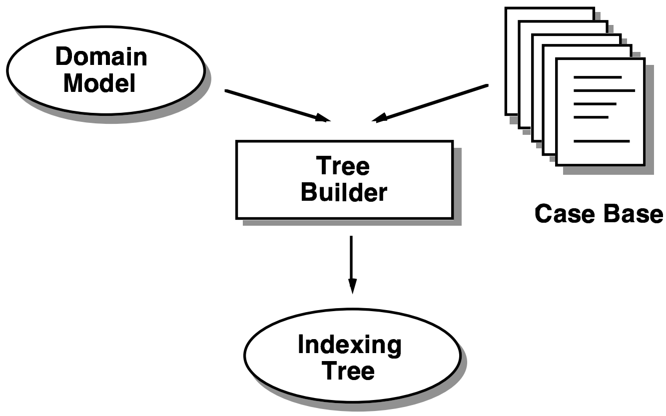

Fig. 3. INRECA System Builder

3 Representing Cases

CASUEL is the common case representation language for the INRECA system and is intended to become the starting point from which a more general European standard case format will be developed. CASUEL is the interface language between all the INRECA component systems, but it is also intended to serve (1) as the interface language between the INRECA integrated system and the external world, and (2) as a standard for exchanging information between classification and diagnostic systems that use cases. Currently, CASUEL is not designed to handle design and planning tasks.

CASUEL is a flexible, object-oriented frame-like language for storing and exchanging descriptive models and case libraries in ASCII text files. It is designed to model naturally the complexities of real cases. CASUEL represents domain objects in a class hierarchy using inheritance, the slots used to describe the objects, type-related constraints on slot values, as well as different kinds of relationships between objects. In its current version, CASUEL additionally supports a rule formalism for exchanging case completion rules and case adaptation rules, as well as a mechanism for defining similarity measures.

4 Representing General Domain Knowledge

When problems are solved through reasoning from cases, the primary kind of knowledge is contained in the specific cases which are stored in the case base. However, in many situations additional general knowledge is required to cope with the requirements of an application. In INRECA, such general knowledge is integrated into the reasoning process in a way that it complements the knowledge contained in the cases (cf. Bergmann, Wess et al., 1994). This general knowledge itself is not sufficient to perform any kind of model-based problem solving (Bergmann, Pews & Wilke, 1994; Pews & Wess, 1993), but it is required to interpret the available cases appropriately.

General knowledge is expressed by three different kinds of rules:

- *Exclusion rules* are entered by the user during consultation and describe hard constraints (knock-out criteria) on the cases being retrieved.
- *Completion rules* are given by the knowledge engineer during the development of the domain model. They describe how to infer additional features out of known features of an old case or the current query case.
- *Adaptation rules* are given by the knowledge engineer during the development of the domain model. They describe how a retrieved case can be adapted to fit the current query.

5 An Example

We now give a short example to illustrate the use of CASUEL. We take the travel agency domain (cf. Lenz, 1993) as an exemplary domain because it is easy to understand. In the following, we give an example of a completion rule and an adaptation rule in CASUEL syntax. For this example, let us assume that the domain model contains (only) three classes: a class *vacation* , a class *transportation,* and a class *accommodation*.. In this model, the classes *transportation* and *accommodation* describe objects which are directly related to the vacation object as shown in Fig. 4. Furthermore, this figure also shows the slots which are used to describe each object. Based on this, a case can be viewed as an instantiation of a vacation, i.e. as an exemplary vacation consisting of (up to) two subobjects (instances of transportation and accommodation, respectively) and some values corresponding to the respective non-relational slots (e.g., duration: 16; price_per_person: 800; number_of_persons: unknown; number_of_adults: 4; number_of_children: 5). The following example is, of course, a rough simplification of the travel agency domain. But, the fully object-oriented capabilities of CASUEL can be used to handle further details like prices for children, prices per season, a hierarchy of accommodations, combinations of accommodations, combinations of transportations etc.

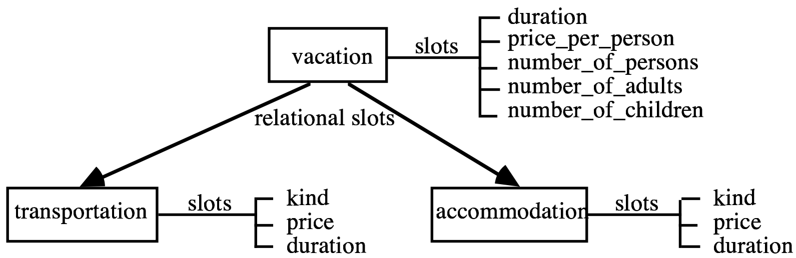

Fig. 4. Exemplary Descriptive Model of the Travel Agency Domain

In CASUEL syntax, the three classes shown in Fig. 4 can be defined as follows:

```
defclass vacation a_kind_of class;
    slots    transportation, accommodation, duration, price_per_person,
             number_of_persons, number_of_adults, number_of_children;
    rules    person_calculation;
    adaptation_rules    price_adaptation.

defclass transportation a_kind_of class;
    slots    kind, price, duration.

defclass accommodation a_kind_of class;
    slots    kind, price, duration.
```

Based on this descriptive model, the following completion rule can be stated to express that *the total number of persons is always the sum of the number of children and the number of adults*.

```
defrule person_calculation of class vacation;
    rule    calculate(add, number_of_adults, number_of_children, ?x) ==>
            number_of_persons := ?x.
```

This rule uses the external function *add* to determine the sum of the two slot values *number_of_adults* and *number_of_children*. The result is assigned to the variable named *?x*. In the conclusion of the rule, the slot *number_of_persons* is assigned the value stored in *?x*.

The adaptation rule for adapting the price of a journey with respect to its duration can be stated as follows:

```
defadaptationrule price_adaptation of vacation;
    rule    query duration > retrieved duration &
            calculate(difference, query duration, retrieved duration, ?additional_days) &
            calculate(multiply, ?additional_days, retrieved accommodation->price,
                                ?additional_price) &
            calculate(add, retrieved price, ?additional_price, ?new_price) ==>
            target price := ?new_price.
```

This adaptation rule first tests whether the duration in the query case is longer than the duration in the retrieved case. Then the first calculate condition computes the difference in the duration. The second calculate condition determines the additional price by multiplying the number of additional days by the accommodation price per day contained in the related object describing the specific accommodation. Finally, the third calculate condition adds the additional price and the price already stated in the retrieved case. In the conclusion of the adaptation rule, the price slot in the target case is assigned the new price.

6 Similarity Assessment

As a consequence of INRECA's fully object-oriented representation, similarity assessment is done on several levels. For each non-relational attribute range-dependent "local" similarity measures are used. To compare two objects, the similarity assessment results with respect to their relational and non-relational slots (i.e. the attribute level) are used to come up with a global similarity. In this view, cases are nothing more than specific objects. Often a weighted sum of the local similarity measures is used as a global assessment mechanism (cf. Tversky, 1977). In INRECA, the seamless integration of induction and case-based reasoning (cf. section 7) allows in addition to extract knowledge inductively and to define improved similarity measures through a learning process (cf. Wess, 1995; Auriol, Wess et al., 1995).

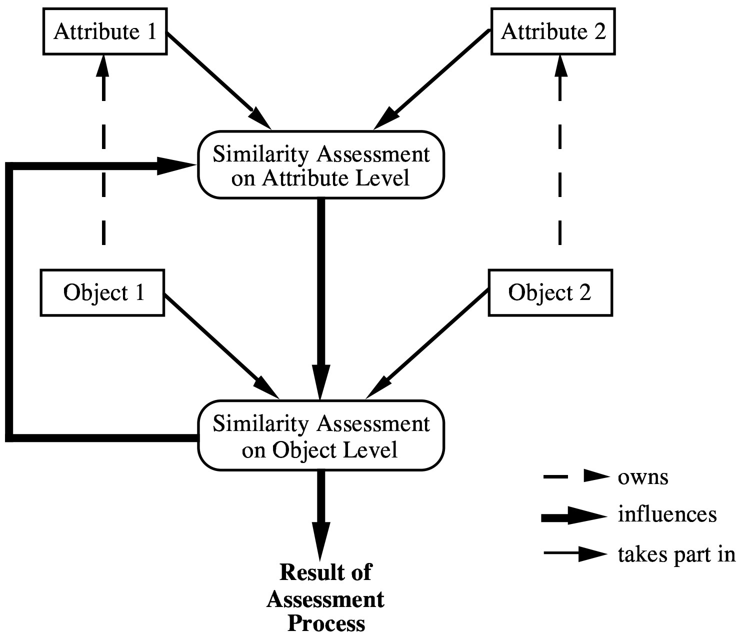

Fig. 5. Similarity Assessment in INRECA (cf. Wess, 1995)

7 Integrating Induction and Case-Based Reasoning

We identified four possible levels of integration between inductive and case-based reasoning technologies. Each level allows several integration possibilities on the different parts of a system presented in the former section. The first level consists simply in keeping both tools as stand-alone systems and letting the user choose the one he is interested in. This toolbox approach should not be rejected because, for instance, a user may feel more comfortable with one method than with another one. In the second level of integration, called co-operative approach, the tools are kept separated but they collaborate: one tool uses the results of the other to improve or speed up its own results, or both methods are used simultaneously to reinforce the results. For instance, the case-based reasoning tool can be used at the end of the decision tree when some uncertainty occurs. In INRECA, communication of results between the tools is achieved through the CASUEL language. The third level of integration, called the workbench approach, goes a step further: the tools are separated but a "pipeline" communication is used to exchange the results of individual modules of each technique. For instance, S^3-CASE produces a similarity measure between a set of cases that may be used by KATE-INDUCTION to supplant the information gain measure. The final level of INRECA reuses the best characteristics of each method to build a powerful integrated mechanism that avoids the weaknesses of each separate technology and preserves their advantages. Fig. 6 summarises the four integration levels.

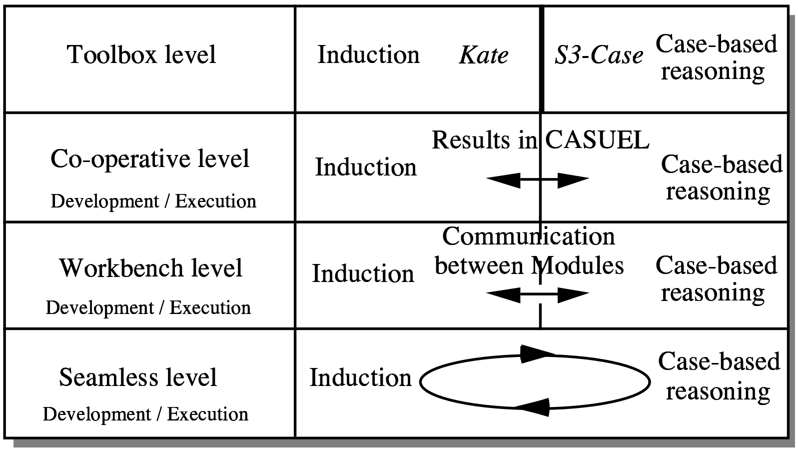

Fig. 6. Four Integration Levels between KATE and S^3-CASE

INTEGRATING KATE-INDUCTION AND S^3-CASE

A very exciting challenge in using together a decision tree and a case-based system is their mutual integration in the execution system. Therefore, the first two levels of integration focus on the consultation part of the system. In the co-operative level, the consultation begins with a decision tree and switches to case-based reasoning when an unknown value is encountered. In the workbench level, the consultation switches in presence of an unknown value between the KATE decision tree and S^3-CASE. This allows to determine the most similar cases to the current situation, to choose the most probable value in this subset of cases and then to switch back to the decision tree. On the other hand, integration can help the development system in providing a better executive output. On the seamless level, specific characteristics of both inductive and case-based techniques are interlaced in a unifying method that allows a (more or less) continuous transfer from the extreme case-based reasoning representation and retrieval standpoint to the induction standpoint, and vice versa.

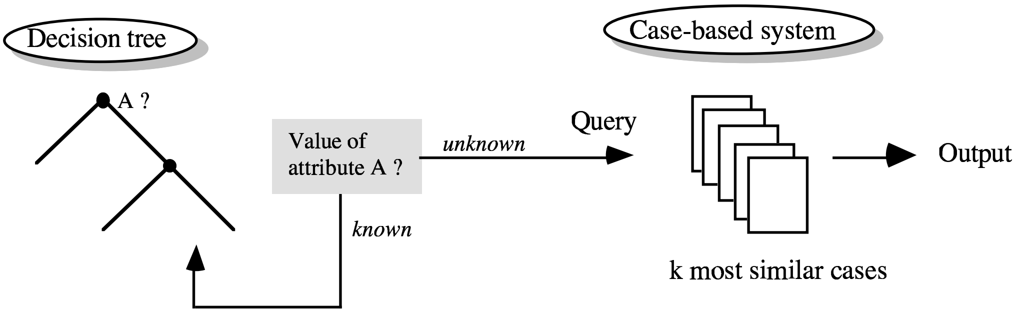

Fig. 7. The Co-operative Level

CO-OPERATIVE LEVEL

The co-operative level aims at switching between the decision tree and the case-based system when facing an unknown value in consultation phase. Each time the decision tree consultation system cannot answer a question (the "unknown values problem", cf. Manago et al., 1993), it switches to the case-based system with the current situation as a query. The case-based system finds the most similar cases and delivers the most probable diagnosis among them (Fig. 7).

WORKBENCH LEVEL

The workbench level defines a double switch between the decision tree and the case-based system. Given a, the current situation defined in the decision tree until an unknown value is encountered, the case-based system retrieves the k most similar cases and "induces" among these cases the most probable value for the current attribute. Then the decision tree can continue its diagnosis further by using this answer. Several links can be used consecutively during the same consultation, what gives its name to this "workbench". The retrieval mechanism is the same as the one presented earlier. The following Fig. 8 sketches the relations between the techniques at this level of integration.

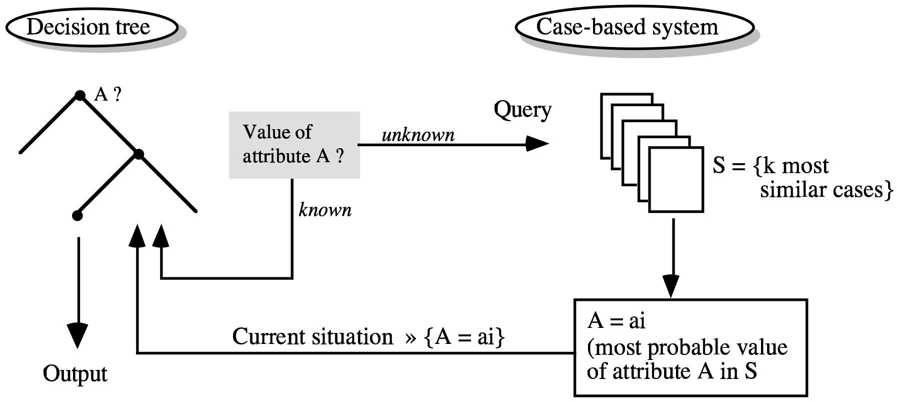

Fig. 8 The Workbench Level

SEAMLESS LEVEL

The most interesting part of the KATE technology to be used in case-based reasoning seems to be the information gain measure (based on Shannon's entropy, cf. Shannon & Weaver, 1947). Information gain is a heuristic that allows the most discriminating attributes to be selected for a given target attribute, such that the resulting tree is minimal in some sense (on average, a few questions are asked to reach a conclusion).

The similarity measure is the basis of the process in a case-based reasoning system. A lot of attention has been paid in PATDEX/S^3-CASE to the definition of the similarity measure such that various aspects that are usually neglected in the classical distance measures (Euclidean, χ^2, etc.) are taken into account here (cf. Althoff & Wess, 1991). To summarise these advantages, one can say that the similarity measure allows the system to be flexible and incremental in a clever way.

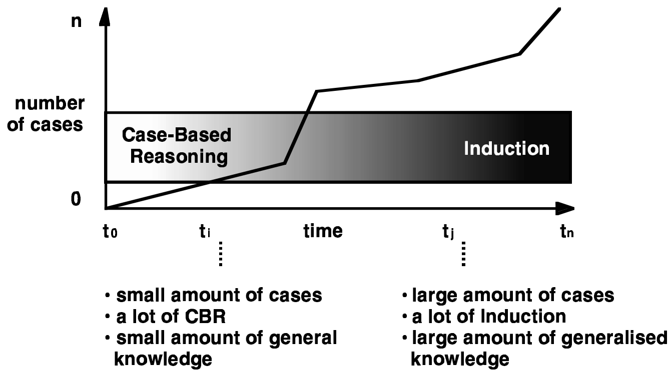

Fig. 9 INRECA System Evolution

On the seamless level, we have one tree with associated generation and retrieval algorithms that enables the combination of the information gain measure for estimating the difference between cases that belong to different classes and the similarity measure for estimating the cohesion of a set of cases that belong to the same class. The INRECA tree (an extension and modification of a k-d tree), with its associated algorithms, can be configured to be a pure decision tree or a pure indexing tree. The INRECA context mechanism in addition allows to do this concurrently, and by this enables the above described integration possibilities based on the seamlessly integrated system (cf. section 8). A learning mechanism automatically extracts knowledge about the similarity of cases from a decision-tree-like INRECA tree and embeds such knowledge into the underlying similarity measure (Auriol, Wess et al., 1995). By this learning method (that is comparable to PATDEX's adaptation method for its similarity measure, cf. Wess, 1991; Richter, 1992; Althoff, Wess et al., 1994), INRECA's k-dimensional indexing tree gradually evolves into a decision tree. Thus, on the seamless level INRECA offers a system evolution over time from pure case-based reasoning to pure induction based on a concept learning strategy (cf. Fig. 9). The theoretical background for this process can, e.g., be found in Wess & Globig (1994a), Wess & Globig (1994b), Wess (1995), and Globig (1995).

8 Applications

Two kinds of vertical systems are built with INRECA, namely in aircraft maintenance (Manago & Auriol, 1995) and in forestry management (Breen, 1994). In addition, INRECA was used as a test environment for a study on reuse of object-oriented software (Bergmann & Eisenecker, 1995).[5]

The databases we use cover a wide range of application domains: the number of cases varies from 125 to 1470, the type of attributes are various (numeric as well as symbolic values), the domain knowledge is more or less important. The following table summarises these databases.

Domain Name	SPONGES	DEVELOPER	TRAVEL	AIRCRAFT	CNC[6]	CAR
# of cases	125	280	1470	621	311	205
# of attributes	35	10	9	7	79	26
# numeric attributes	10	3	3	0	0	14
# values per attribute (average)	4	18	26	33	11	6
% unknown values (average)	37	26	0	17	94	1
Database characteristics	complex case structure	no fixed outcome, high number of unknowns	no fixed outcome, no unknown values	high number of unknowns and values per attribute	very high number of unknown values	well-balanced domain

The "SPONGES" domain deals with the recognition of the specie of a marine sponge on the basis of a structured sponge description. This application has been developed within the Museum of Natural History (Paris, France). In the "DEVELOPER" domain, one has to decide which kind of chemical is necessary when knowing the development film, the current temperature, etc. These data have been provided by tecInno. The "TRAVEL" domain is a classical problem that arises in many travel agencies: given some wishes of the traveller (price, destination, type of holidays, etc.), the agency has to find an adequate hotel. The data on the "Travel Agency" domain were provided by Mario Lenz to whom we are indebted. The "AIRCRAFT" domain deals with the maintenance of planes' engines. The database, that is a subset of a real application dataset, has been provided by AcknoSoft. The "CNC" domain deals with failure diagnosis in a computerised numeric control machine. These data have been provided by the University of Kaiserslautern and tecInno. They are based on a co-operation with the chair for metrology and quality management in production at the Technical University of Aachen (Pfeifer & Faupel, 1993; Althoff, Faupel et al., 1989). In the "CAR" domain, one has to determine a risk estimation of a car for insurance companies on the basis of several attributes such as manufacturing, price, technical data, etc. The "CAR" domain comes from the UCI Repository of Machine Learning Data Bases and Domain Theories, USA.

FEEDBACK FROM APPLICATIONS

The INRECA architecture has been influenced by the experience the University of Kaiserslautern and tecInno have made with the development of the MOLTKE workbench in general (Althoff, 1992a; 1992b;

---

5 Also other application tasks have been analysed (e.g., Traphöner & Althoff, 1993).

6 CNC = Computerized Numerical Control

Pfeifer & Richter, 1993) and the PATDEX system in particular (Wess, 1991; Wess, 1993; Althoff & Wess, 1991; Richter & Wess, 1991; Althoff, 1993). While the MOLTKE family of systems was designed to deal with technical diagnosis problems (Pfeifer & Faupel, 1993), the integrated INRECA system was opened up to deal with decision support problems in general. There are three main sets of problems where INRECA is designed for, namely:

- *Classification*
  The requested output is one or several classes out of a fixed set of possible ones.
- *Diagnosis*
  In addition to the above classification problem, the problem of acquiring additional information (test selection problem) has to be solved.
- *Decision support*
  The requested output is dynamically defined. In principle, it can be any combination of attributes that are not specified in the query.

The complexity of dealing with general decision support domains that have dynamically changing targets was first investigated using the DEVELOPER domain. While the inductive learning strategy being applied on the level of seamless integration is well suited for classification and diagnosis problems (it needs explicitly defined classes), it requires an additional focusing mechanism for decision support problems that have no fixed target. For enabling learning to improve the overall system performance, we introduced the concept of a context. In the underlying idea, it is similar to the context concept used in the MOLTKE workbench, but it can also be seen as a dynamic extension of the view concept known from databases.

Contexts can be statically defined in the domain model, or dynamically by the user for structuring the user domain w.r.t. the underlying important assumptions that influence the solution. Using contexts allows the realisation of all levels of integration based on the seamlessly integrated system. This can behave like a pure decision tree or like a pure case-based reasoning system. Via the context mechanism several instances of the seamlessly integrated system are used in parallel. The concept of a context is an extension of the prototype and template concept used in ReMind (e.g., Barletta, 1994). Basic parts of the context mechanism are already implemented. The completion is currently under development.

As it is stated in Manago & Auriol (1995), the experience with the AIRCRAFT domain showed that the seamlessly integrated system needs further improvement in terms of performance. Here we introduced domain-dependent user-selectable search heuristics (e.g., automatic application of tests whether to stop retrieval, or not) that significantly improved the performance. This also underlined the need of a context mechanism.

One important result from using so different application domains was that we need a combination of different retrieval strategies for achieving the necessary flexibility. For satisfying given domain and/or user constraints we needed a combination of retrieval based on strict queries and similarity-based retrieval, in the SOFTWARE REUSE domain we simply applied sequential retrieval because of the underlying case complexity, and in the CAR and the FORESTRY MANAGEMENT domain the experiments carried out showed that the seamlessly integrated system is the best solution here because of both in terms of performance and of classification accuracy (cf. Auriol, Wess et al., 1995).

9 Evaluation

We evaluated several commercially available case-based reasoning tools, namely KATE tools, CBR EXPRESS, ESTEEM, REMIND, and S^3-CASE[7] and discussed them from a pragmatic perspective (cf. Althoff, Auriol et al., 1995). The reasons for conducting this evaluation were to get deeper "insights" for the final INRECA system to come up with a reasonable and realistic combination of research and application issues that should be solved.

The main novelty of our evaluation is that a set of experiments have been conducted with the tools using the same case data. In order to conduct these experiments, we had to define a set of objective evaluation criteria (mostly specialised to case-based reasoning but also including some more general aspects where useful; cf. Traphöner, Manago et al., 1992; Althoff, Auriol et al., 1995) and design test procedures in order to apply these evaluation criteria onto the systems. We have chosen case databases that demonstrate different features and allow to test different functionalities and properties of the systems. From conducting all these experiments with tools currently available on the market, we gained valuable feedback for guiding our research efforts on the INRECA system, especially on the integration of induction and case-based reasoning.

---

[7] Two other vendors had been asked to participate in this evaluation but did not support us with an evaluation copy.

We carried out exemplary implementations of four test domains for the respective tools. Every tool was intensively tested by one team member at the University of Kaiserslautern. There have been two main criteria for selecting the test domains: the necessity to have enough competence within the evaluation team, and to have at least one domain that should be easily understandable by anyone. We chose the "Fault Diagnosis of CNC Machining Centres" domain, a domain where the university group has many years of experience, the "Classification of MARINE SPONGES" domain where we had the opportunity to involve a real expert, and the TRAVEL AGENCY domain that meets the requirements of being easily understandable. In addition, we chose the CAR domain because it is well known.

10 Summary and Conclusion

The integration issue appears to be the most interesting result from our evaluation. The more complex real world applications are, the higher is the need for having deeply integrated systems. Current industrial case-based reasoning tools are at the toolbox level only (with minor exceptions). INRECA's seamless integration of inductive and case-based reasoning allows to come up with a decision support system that includes a well-suited combination of both technologies. INRECA keeps the classification quality of case-based reasoning and its incrementallity, but it can also gain speed by inductively learned explicit knowledge. Depending on the application task at hand, INRECA's problem solving method can be flexible or straight forward. As a consequence of this, INRECA enables the building of decision trees that are more resistant to noise and the building of more efficient indexing structures for case-based reasoning.

The main goal of the INRECA project is making tools and methods available for developing, validating, and maintaining decision support systems with a special focus on the integration of inductive and case-based technologies. From a long-term perspective, this can be seen as going first steps towards the development of future AI systems, or of software systems in general, that have adaptive capabilities, i.e. that are able to learn from their own problem solving experiences. Thus, the INRECA system, as a seamless integration of inductive and case-based reasoning technology in the form of KATE-INDUCTION and S^3-CASE, contributes to the current state of technology.

11 Acknowledgement

Funding for INRECA has been provided by the Commission of the European Communities (ESPRIT contract P6322) to which the authors are greatly indebted. The partners of INRECA are AcknoSoft (prime contractor, France), tecInno GmbH (Germany), Interactive Multimedia Systems (Ireland), the University of Kaiserslautern (Germany). KATE is a trademark of Michel Manago. S^3-CASE is a product of tecInno.

12 Literature

Aha, D. (ed.) (1994). *Proc. AAAI-94 Workshop on Case-Based Reasoning*, AAAI Press

Althoff, K.-D. (1992a). *Eine fallbasierte Lernkomponente als integrierter Bestandteil der MOLTKE-Werkbank zur Diagnose technischer Systeme*. Doctoral Dissertation, University of Kaiserslautern; also: Sankt Augustin, infix Verlag

Althoff, K.-D. (1992b). Machine Learning and Knowledge Acquisition in a Computational Architecture for Fault Diagnosis in Engineering Systems. In: M. Weintraub (ed.), *Proc. ML92 Workshop on "Computational Architectures for Supporting ML and KA" in Aberdeen*

Althoff, K.-D. (1993). Lernverfahren in MOLTKE In: *Pfeifer & Richter (1993)*, 173-200

Althoff, K.-D., Auriol, E., Barletta, R., Manago, M. (1995). *A Review of Industrial Case-Based Reasoning Tools*. AI Perspectives Report, Oxford, UK: AI Intelligence

Althoff, K.-D., Bergmann, R., Maurer, F., Wess, S., Manago, M., Auriol, E., Conruyt, N., Traphöner, R., Bräuer, M. & Dittrich, S. (1993). Integrating Inductive and Case-Based Technologies for Classification and Diagnostic Reasoning. In: E. Plaza (ed.), *Proc. ECML-93 Workshop on "Integrated Learning Architectures"*

Althoff, K.-D., Faupel, B., Kockskämper, S., Traphöner, R. & Wernicke, W. (1989). Knowledge Acquisition in the Domain of CNC Machining Centers: the MOLTKE Approach. In: J. Boose, B. Gaines & J.-G. Ganascia (eds.), *Proc. EKAW-89*, 180-195

Althoff, K.-D. & Wess, S. (1991). Case-Based Knowledge Acquisition, Learning, and Problem Solving in Diagnostic Real World Tasks. *Proc. EKAW-91*, Glasgow & Crieff

Althoff, K.-D., Wess, S., Bergmann, R., Maurer, F., Manago, M., Auriol, E., Conruyt, N., Traphöner, R., Bräuer, M. & Dittrich, S. (1994). Induction and Case-Based Reasoning for Classification Tasks. In: H. H. Bock, W. Lenski & M. M. Richter (eds.), *Information Systems and Data Analysis, Prospects-Foundations-Applications,* Springer Verlag, Berlin-Heidelberg, 1994, 3-16

Auriol, E., Manago, M., Althoff, K.-D., Wess, S. & Dittrich, S. (1994). Integrating Induction and Case-Based Reasoning: Methodological Approach and First Evaluations. In: *Keane, Haton & Manago (1994),* 145-156

Auriol, E., Wess, S., Manago, M., Althoff, K.-D. & Traphöner, R. (1995). On the Integration between Induction and Case-Based Reasoning: Theoretical Advances and Case Studies. *Submitted*

Barletta, R. (1994). A Hybrid Indexing and Retrieval Strategy for Advisory CBR Systems Built with ReMind. In: Keane, Haton & Manago (1994), 49-58

Bartsch-Spörl, B., Janetzko, D. & Wess, S. (eds.) (1995). *Fallbasiertes Schließen - Grundlagen und Anwendungen.*. Proc. of the 3rd German Workshop on Case-Based Reasoning (at the XPS-95 conference, organised by AK-CBR), LSA-REPORT 95-02, University of Kaiserslautern

Bergmann, R. & Eisenecker, U. (1994). Fallbasiertes Schließen zur Unterstützung der Wiederverwendung objektorientierter Software: Eine Fallstudie. In: M. M. Richter & F. Maurer (eds.), *Expertensysteme 95, Proc. 3rd German Expert System Conference*

Bergmann, R., Wess, S., Traphöner, R. & Breen, S. (1994). Using Background Knowledge in the Integrated System *INRECA Deliverable D29, Version 1*

Bergmann, R., Pews, G. & Wilke, W. (1994). Explanation-Based Similarity: A Unifying Approach for Integrating Domain Knowledge into Case-Based Reasoning for Diagnosis and Planning Tasks. *Wess, Althoff & Richter (1994)*, 182-196

Breen, S. (1994). Forestry Management Using Case-Based Reasoning. *Talk at EWCBR-94 Industry Day*

Globig, C. (1995). Fallbasiertes Repräsentieren und Lernen von Begriffsmengen. In: *Bartsch-Spörl, Janetzko & Wess (1995)*

Keane, M., Haton, J. P. & Manago, M. (eds.) (1994). *EWCBR-94 - Second European Zorkshop on Case-Based Reasoning*. Paris: AcknoSoft Press

Lenz, M. (1993). CABATA - a hybrid CBR system. In: *Richter, Wess et al. (1993)*, 204-209

Manago M. (1989). Knowledge Intensive Induction. *Proc. of the sixth International Machine Learning workshop, Morgan Kaufmann*

Manago M. (1990). KATE: A Piece of Computer Aided Knowledge Engineering. *Proc. of the 5th AAAI workshop on KA for knowledge based systems,* Gaines B. & Boose J. eds., Banff, AAAI Press

Manago, M., Althoff, K.-D., Auriol, E., Traphöner, R., Wess, S., Conruyt, N., Maurer, F. (1993). Induction and Reasoning from Cases. In: *Richter, Wess et al. (1993)*, 313-318

Manago, M, Althoff, K.-D. & Wess, S. (1994). Comparison of Induction and Reasoning from Cases. In: *Reasoning with Cases: Theory and Practice*, Supporting Material for ECAI-94 Tutorial

Manago, M., & Auriol, E. (1995). Integrating Induction and Case-Based Reasoning for Troubleshooting CFM-56 Aircraft Engines. In: *Bartsch-Spörl, Janetzko & Wess (1995)*

Manago, M., Bergmann, R., Conruyt, N., Traphöner, R. Pasley, J., Le Renard, J., Maurer, F., Wess, S., Althoff, K.-D. & Garry, S. (1994). CASUEL: A Common Case Representation Language. *INRECA Deliverable D1, Version 2.01*

Pews, G. & Wess, S. (1993). Combining Case-Based and Model-Based Approaches for Diagnostic Applications in Technical Domains. In: *Richter, Wess et al. (1993),* 325-328

Pfeifer, T. & Faupel, B. (1993). Die Anwendung von MOLTKE: Diagnose von CNC-Bearbeitungszentren. In: *Pfeifer & Richter (1993)*, 42-67

Pfeifer, T. & Richter, M. M. (eds.) (1993). *Diagnose von technischen Systemen - Grundlagen, Methoden und Perspektiven der Fehlerdiagnose*. Deutscher Universitäts-Verlag.

Richter, M. M. (1992). Classification and Learning of Similarity Measures. *Proc. 16th Annual Conference of the German Society for Classification*, Springer Verlag

Richter, M. M. & Wess, S. (1991). Similarity, Uncertainty and Case-Based Reasoning in PATDEX. *Automated Reasoning - Essays in Honor of Woody Bledsoe*, Kluwer Academic Publishers.

Richter, M. M., Wess, S., Althoff, K.-D. & Maurer, F. (eds.) (1993). *Proc. 1st European Workshop on Case-Based Reasoning (EWCBR-93)*. SEKI-REPORT SR-93-12, University of Kaiserslautern

Shannon & Weaver (1947). *The Mathematical Theory of Computation*. University of Illinois Press

Traphöner, R. & Althoff, K.-D. (1993). Recent developments in case-based reasoning and inductive learning and their applicability in automotive industries. In: *Proc. 26th International Symposium on Automotive Technology and Automation, Dedicated Conf. on Mechatronics*, Aachen, Germany

Traphöner, R., Manago, M., Conruyt, N., Dittrich, S. (1992). Industrial Criteria for Comparing Technologies in INRECA. *INRECA Deliverable D4, Version 1*

Tversky, A. (1977). Features of Similarity. *Psychological Review **84***, 327-352

Wess, S. (1991). *PATDEX/2: ein System zum adaptiven, fallfokussierenden Lernen in technischen Diagnosesituationen*. SEKI Working Paper SWP-91-01, University of Kaiserslautern, 1991

Wess, S. (1995). *Fallbasiertes Schließen in wissensbasierten Systemen zur Entscheidungsunterstützung und Diagnose*. Doctoral Dissertation, University of Kaiserslautern (forthcoming)

Wess, S., Althoff, K.-D. & Derwand, G. (1993). Improving the Retrieval Step in Case-Based Reasoning. In: *Richter, Wess et al. (1993)*, 83-88

Wess, S., Althoff, K.-D. & Derwand, G. (1994). Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning. In: *Wess, Althoff & Richter (1994)*, 167-181

Wess, S., Althoff, K.-D. & Richter, M. M. (eds.) (1994). *Topics in Case-Based Reasoning*. Springer-Verlag

Wess, S. & Globig, C. (1994a). A Case Study on Case-Based and Symbolic Learning. In: *Aha (1994)*

Wess, S. & Globig, C. (1994b). Case-Based and Symbolic Classification Algorithms - A Case Study Using Version Space. In: *Wess, Althoff & Richter (1994)*, 77-91