The Development of HOMER
A Case-Based CAD/CAM Help-Desk Support Tool

Mehmet Göker[1], Thomas Roth-Berghofer[2],
Ralph Bergmann[3], Thomas Pantleon[4], Ralph Traphöner[2],
Stefan Wess[2], Wolfgang Wilke[3]

[1] Daimler-Benz Research and Technology 3, FT3/KL
P.O. Box 2360, D89013 Ulm, Germany
mehmet.goker@dbag.ulm.daimlerbenz.com
Phone: +49 731 5052856 Fax: +49 731 5054210

[2] tecInno GmbH
Sauerwiesen 2, D67661 Kaiserslautern, Germany
{roth, trappi, wess}@tecmath.de
Phone: +49 6031 60668 Fax: +49 6031 60667

[3] University of Kaiserslautern LSA
P.O. Box 3049. D-67653 Kaiserslautern, Germany
{bergmann, wilke}@informatik.uni-kl.de
Phone: +49 631 2053361 Fax: +49 631 2053357

[4] Daimler-Benz Personal Car Development EP/QDF
Hauspostcode G270, D-71059, Sindelfingen, Germany
Thomas.Pantleon@PCD.Daimler-Benz.com
Phone: +49 7031 909574 Fax: +40 7031 909885

Abstract. The increasing number of hardware and software at Daimler-Benz personal car development in Sindelfingen combined with the constant number of help-desk operators demanded a help-desk system which goes beyond the classical trouble-ticket approach. In this application paper we give an overview of the situation at the CAD/CAM Help-Desk in Sindelfingen and the development of the case-based help-desk support tool HOMER. We describe our modeling approach and its influence on the system architecture as well as the different user roles and the help-desk tool itself. We conclude with the lessons learned during the course of this project and future prospects.

1 Motivation

The CAD/CAM help-desk at Daimler-Benz in Sindelfingen (EP/QDF) provides support for the engineers developing Mercedes-Benz cars. Currently approximately 1000 engineers work on workstations utilising 20 different applications in average.

The number of hardware and software is going to increase drastically during the next years. Also, each year one third of the hardware is replaced with newer models and each software has about one major and up to four minor updates. Since the number of help-desk personnel is going to remain constant and the help-desk operation is

a central and very time-critical operation, it is obvious that some kind of computer-support that goes beyond the classical trouble-ticketing systems is needed.

2 Current Situation at the CAD/CAM Help-Desk

Currently the Help-Desk in Sindelfingen is organised in three layers (Figure 1). The end-users are the engineers that are working on the development of the Mercedes-Benz cars in Sindelfingen. When the engineers have a problem with their software or hardware the first person they contact is usually the "key-user". This is an engineer, working in the same group, that has more experience with the software in use and/or the hardware. If the key-user is not able to help, the first level support is contacted. This level is divided into several groups. The system help-desk gives support on operating system and simple hardware problems. The application help-desks are specialised on certain software and can give support on their usage and functionality. If the first level help-desk cannot solve the problem, it is escalated to the second level, i.e. the system administrators and application specialists. The personnel at this level is

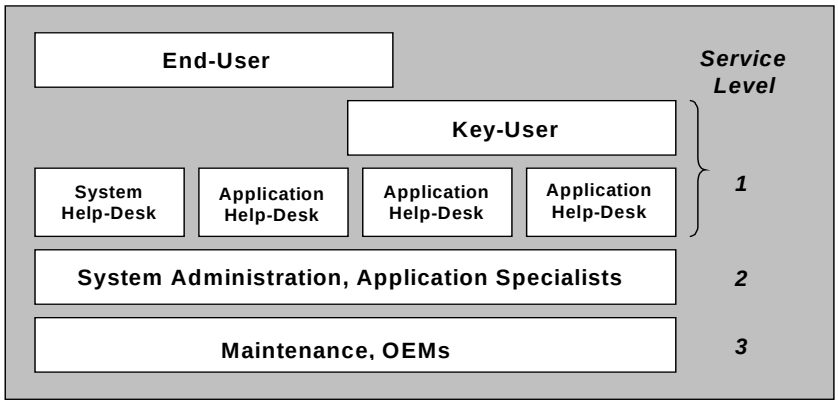

Figure 1: Support Layers at the CAD/CAM Help-Desk

comprised of highly skilled and specialised system administrators. Problems are transferred to specific system administrators based on their area of expertise. As the overlap in the areas of expertise between the administrators is rather small, problems have to wait if the required administrator is not available. Problems that cannot be solved by the second level support are transferred to the hardware and software vendors – some of which have representatives in-house.

Currently the operators at the help-desk are using several software tools to support them during help-desk operations (Figure 2). HIT is a "Help-desk Information Tool" that has been developed in-house to give the help-desk operators information about the user and his default environment from the network (NIS). CIVIS is an inventory system, that provides maintenance and hardware configuration information to the help-desk operator. FES is a trouble-ticket tool that is used to record, track and escalate calls.

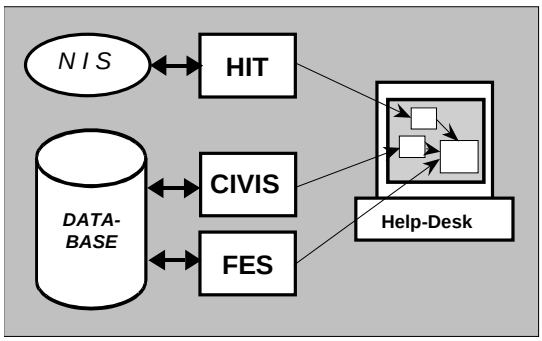

Figure 2 : Current Software Tools used by the Help-Desk Operators

Even though these tools aid the help-desk operators in performing their tasks, they do not give support in diagnosing the problem and cannot serve as a knowledge repository. The growing number of hard- and software as well as the goal to drastically shorten development cycles heavily increase the responsibility of and the pressure on the help-desk.

Although the personnel at the help-desk is very well trained, it is impossible for them to be knowledgeable in all areas. The complexity and the dynamics of the domain make it infeasible for one help-desk operator to understand and manage the entire hardware, software and network. The end-users, i.e. the engineers get frustrated when they have to be transferred from one help-desk operator to the other and the self-esteem of the operator suffers as well.

Since most of the problems recur, the operators end up re-deriving solutions to problems that a colleague of them has already solved in the past. This is not only frustrating to the help-desk operator and the engineer, but also a waste of resources of the company.

3 Project Goals and Expected Benefits

In order to increase the efficiency and effectiveness of the help-desk and thereby increase the productivity of the car development, it is in the interest of Daimler-Benz to cumulate the experience of the help-desk operators in a knowledge repository and make it available to all operators on all levels (Figure 3). In this manner the corporate knowledge, one of the most valuable assets of the company, is going to be cumulated, preserved and extended. This will ensure that solutions to previous problems will be available to all levels at the help-desk, whenever and wherever they are needed, and the training of new help-desk operators will be expedited. With the availability of the cumulated knowledge of the help-desk at the finger-tips of each operator, the quality of the support will increase significantly and the downtime of hard-and software will decrease. The repository can also be used to analyse the recurrence of problems to determine weak spots in hard- and software and to check what one has to be prepared for during updates or migration. Once the experience of the operators has been captured, it can also be transferred between various sites of Daimler-Benz world-wide. This will ensure that the same high standards of quality will be established at every location.

The knowledge repository that contains the experience of the help-desk operators as well as the tools to access this knowledge has to be easy to create, maintain and

update. Consistency, usability and validity are essential. It is impossible to create and maintain a model-based system of such a complex domain. Obviously a rule-based expert system approach is also bound to fail. Since solutions in the form of cases do exist and because of the obvious advantages in terms of maintainability, the case-based reasoning approach was chosen to implement HOMER, the case-based help-desk support tool (*Ho*tline *m*it *Er*fahrung – Hotline with experience). To do this, EP/QDF co-operates with Daimler-Benz Research and Technology (FT3/KL) which in turn participates in the ESPRIT Project INRECA-II[i].

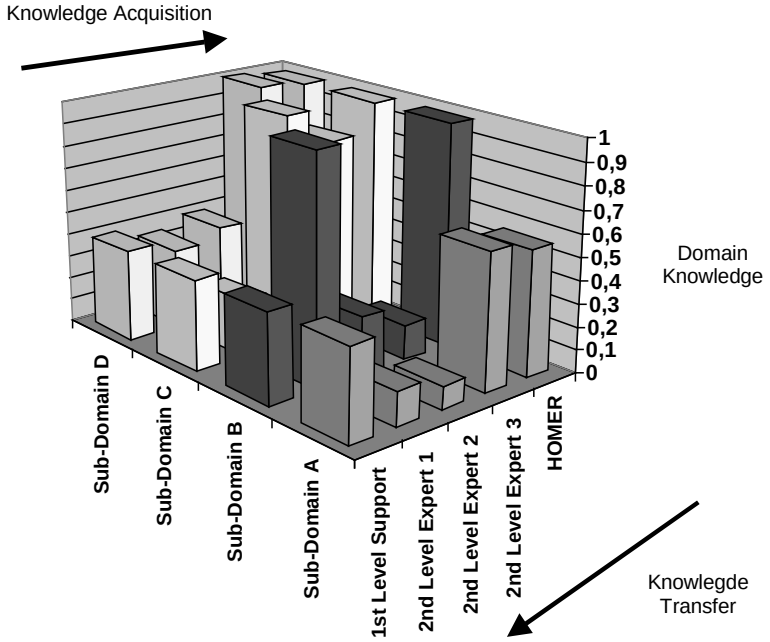

Figure3: Knowledge Acquisition and Transfer

To estimate the financial gains that can be achieved by supporting the help-desk with a case-based system, we took the calls that required more than 10 minutes and less than one day to solve as a basis for our calculation. We assumed, that the operators would not use the system for trivial calls (less than 10 minutes) and that problems that take longer than one day to solve require new hardware or software. Our calculations showed, that the project covered its costs if we managed to reduce the average time needed to answer a call by 50% for 40% of the incoming calls, i.e. 60% of the calls are solved in the same time and 40% of the problems in half the time.

4 Domain Modelling and Case Acquisition

4.1 Domain Model

One of the major challenges one has to face when creating a system to capture and represent the experience of domain experts, is to determine the level of abstraction with which the domain and the knowledge is going to be modelled. If the model used is too simplistic it will cause problems while capturing the experience and will miss important details. If, however the domain model is too specific, one will get quickly lost in useless details, and knowledge acquisition will be very tedious and time-consuming. Maintenance will be very difficult for both a too simplistic and a too complex model.

When we started the development of the system, representing the domain in a flat list of attribute value pairs or putting the experience of the operators in a textual, question-answer oriented case-base seemed to be very tempting [c.f. 1]. Since the modelling effort would be rather low, the operators could enter cases themselves and knowledge acquisition would proceed rather swiftly.

However, we soon realised, that the cost of this 'shallow' approach was the maintenance effort one has to put into the case-base after a certain size has been reached. Since the system has very limited knowledge about the structure and semantics of the domain, it cannot aid in structuring the cases. Similarity calculation in such systems is based purely on surface features, the number of identical answers to questions or textual pattern matching of the cases. Using a flat attribute-value representation, the help-desk operator would eventually be faced with hundreds of attributes from which he/she has to choose from and which have to be specified. To avoid redundant questions and information in a question/answer based textual approach, one has to create a decision tree for the questions by hand. This task is similar to maintaining a rule-base and unfeasible for complex domains and large case-bases.

We decided to use an object-oriented approach to model the domain. While the effort necessary to create such a model is obviously higher, it can be used in guiding the help-desk operator while describing and entering cases, and during similarity calculation. The level of abstraction of the domain model was discussed in detail with the system administrators and a level which they felt comfortable with was selected. Since the object-oriented domain model forms the basis for case representation an additional effort is necessary in the beginning of knowledge acquisition. However, the better the domain is modelled, the easier it is to maintain and use the system afterwards.

The decision to use a domain model approach opposed to a shallow approach depends also on the intended users of the system. For inexperienced help-desk operators a tool with which simple problems can be solved by answering a limited number of questions is of great value [1]. However, in our case, the intended users are experienced help-desk operators who would not bother to use a system for (subjectively) trivial problems. In such an environment, a 'shallow' system is of limited use. The

system has to be able to present the not-so-obviously similar solutions that the help-desk operators could not find. Since knowledge contained in the domain model is used in similarity calculation, the retrieved solutions are similar in a semantic and structural manner. The domain model allows the solutions in the case-base to be applicable to a broader range of problems.

4.2 Case Model

The cases in the help-desk domain were modelled in accordance with the approach the help-desk operators use in solving problems (Figure 4).

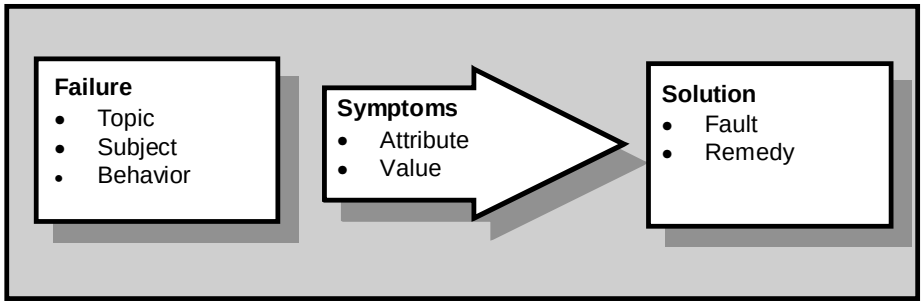

Figure 4: Basic structure of a Help-Desk Case

The first thing a help-desk operator has, is the *Failure* description given by an end user. This description is a subjective description and may or may not have something to do with the actual cause of the problem. The *Failure* in the case model comprises

- the *Topic*: the area in which the problem is located (hardware, software, network, printing etc.),
- the *Subject*: the physical object that the failure is related to (specific software, printer, screen etc.)
- the *Behaviour*: the way the subject (miss-) behaves (crashes, wrong print size, screeching sound etc.)

The *Symptoms* are a list of attribute-value pairs by means of which the help-desk operator is able to diagnose the fault. The symptoms contain the minimum amount of information that is necessary to diagnose the problem.

The *Diagnosis* contains the 'Fault', i.e. what the cause of the problem was, and the 'Remedy', i.e. how to solve the problem. In the actual implementation of the system there is also some administrative information regarding the specifications needed to apply this remedy and how long it takes to solve the problem.

One has to bear in mind, that each failure description can be the result of various symptoms which in turn can be the result of various faults which can again be solved by applying various remedies. Each complete path from a failure to a solution is a case in our system.

4.3 Influence of the Domain Model on System Design

To ensure that every help-desk operator accesses the same, up-to-date case-base from every point in the network, the system is implemented using a client-server architecture. This enables us to use one central domain model and case-base, and the domain model and case-base are also easier to maintain.

Figure 5 shows the main components of the system. The CBRWorks Server is accessed through the high bandwidth intranet of Daimler-Benz by the *HOMER Clients*. In terms of client-server systems the *HOMER Client* is a *fat* client. It contains the whole domain model which is loaded on client start-up. It therefore can build up que-

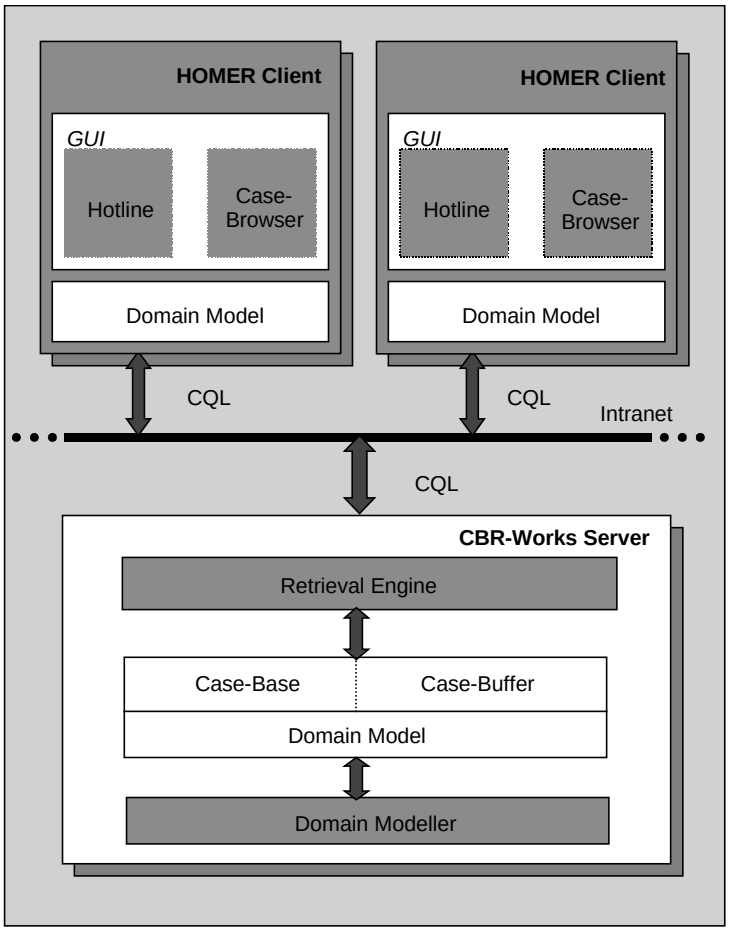

Figure 5: System Architecture

ries and cases based on the domain model and sends requests to the server only when needed. This obviously reduces network traffic. The communications language used is CQL (case query language) which was derived from CASUEL, a query language developed in the INRECA project [2].

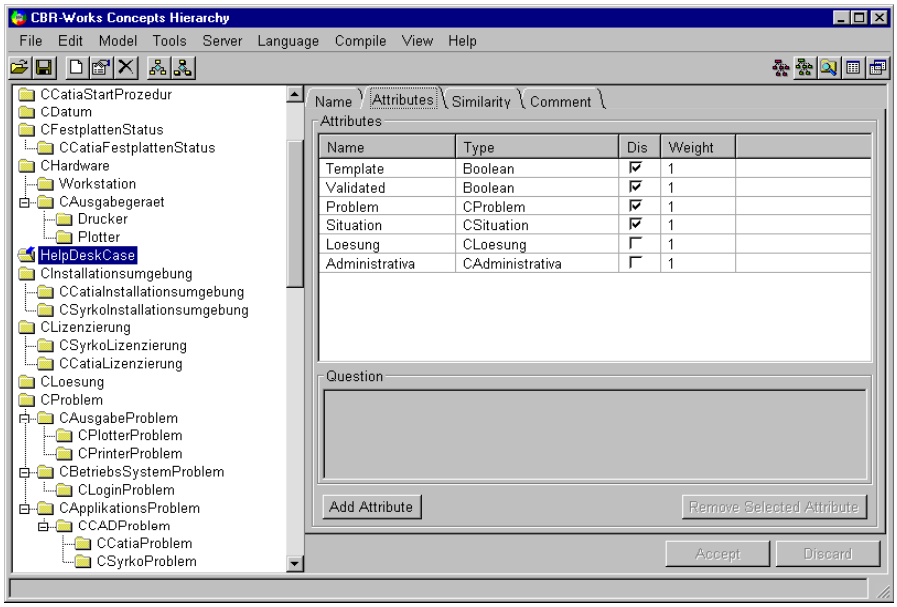

Figure 6: Domain Modelling

CBR-Works Server is a tool developed by tecInno. Since it is written in Smalltalk it is hardware independent. The system at Daimler-Benz is currently running on a Windows NT machine. The *CBR-Works Server* provides the model and the case-base as well as the tools that the administrator needs to model and maintain the domain.
Figure 6 shows a snapshot of a small part of the domain model of HOMER. On the left hand side a tree view displays the hierarchical structure of the domain concepts. The *HelpDeskCase* is selected to show the attribute slots.

The slot *Problem* contains a concept *CProblem* which describes the failure as mentioned above. *Situation* describes the symptoms. These are structured in subconcepts to ease the maintainability of the domain model and to speed up the retrieval process. *Loesung* holds the solution to the given problem and *Administrativa* stores information about the person who stored this case into the case-base and the like. *Loesung* and *Administrativa* are not *Discriminant*, i.e. these slots are not taken into account during similarity calculation. The slots *Validated* and *Template* are boolean flags that are used internally by the system.

The cases are stored in two different areas, in the *case-buffer* and the actual *case-base* (Figure 5). The case-buffer is used to store the newly created cases which have not been approved yet by the Case-Base Editor (Chapter 5.1). The validated cases reside in the case-base. The help-desk operator has the choice to restrict the search to the approved cases or to make use of the case-buffer as well.

The *HOMER-Client* was written entirely in Java. Earlier versions have been developed as HTML-Pages and later with JavaScript but the dynamics and complexity of the domain model made development with these tools infeasible. Java was chosen because of its networking capabilities and the ease of making it available to the help-

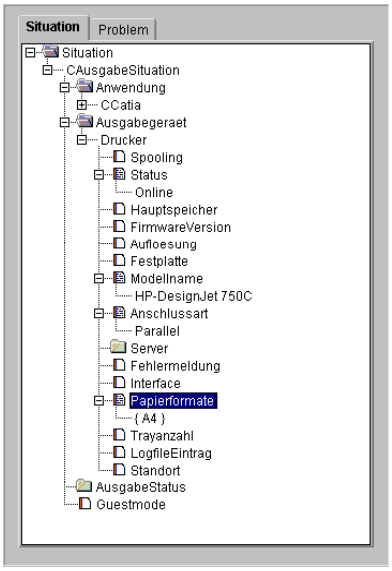

Figure 7: Query building in user driven mode

desk operator through a web server. The fast intranet allowed for a relatively big Applet, such that a fat client implementation was possible.

The help-desk tool assists the operator with two modes of execution. In a *user driven mode* the operator can build up a query based on the domain model (Figure 7). This mode is meant for the experienced help-desk operator who does not need guidance in what to ask, knows the case structure and wants to enter the data directly.

For the operator with less experience or if the operator wants to have support in what to ask next, he/she can switch to the *system driven mode*. In this mode the help-desk tool presents the most relevant questions to the operator. The questions are generated using an entropy measure which selects the attributes with the highest information gain based on the current case-base. The help-desk operator can choose which question to ask the end user from the list of these questions

5 User Types and the User Interface

5.1 User and Roles

The complexity and size of the domain as well as the demanded accuracy and consistency of the captured and re-used experience require the implementation of processes for case-modelling, case-acquisition, case-maintenance and case-review.

As already mentioned above, the case-base is separated into a case-buffer and a main case-base. The main case-base contains reviewed, prototypical cases. The experience in the main case-base represents the approved way of best practice for the CAD/CAM help-desk. The case-buffer contains the cases that have been created by the help-desk operators during their daily work. They may or may not be worth transferring to the main case-base. Since they do contain information that is relevant for the daily operation of the help-desk, they are available to all help-desk operators but marked as being 'not approved'.

To implement the necessary processes to operate HOMER, three user types have been defined: the System Administrator, the Case-Base Editor and the Help-Desk Operator.

The Help-Desk Operator has the lowest access rights in the system. He/she is the person that uses HOMER on a daily basis to solve the problems of the end-users. The main tasks of the help-desk operator are case-acquisition and case-retrieval.

The Case-Base Editor is responsible for case maintenance and case approval. He/she checks the cases in the case buffer and transfers relevant cases to the main case-base. The case-base editor also has the duty to check for redundancy and consistency in the case-base. He/she may modify the value ranges of the attributes in the domain model but is not allowed to modify the domain model itself (i.e. add/remove attributes, move concepts etc.).

The System Administrator has the highest access rights. He/she has to create and maintain the domain and case model for the system, and administer the users and their access rights.

5.2 User Interface

The Help-desk tool gives the operator all relevant information he/she needs in an easily understandable manner, at one glance (Figure 8). On the left hand side, a tree view shows an actual query situation represented in the domain model. This tree view can be switched off if the operator wants to go into the system driven mode.

The list on the upper right hand side displays already answered questions and the corresponding values in the most similar cases. The second list on the right hand side

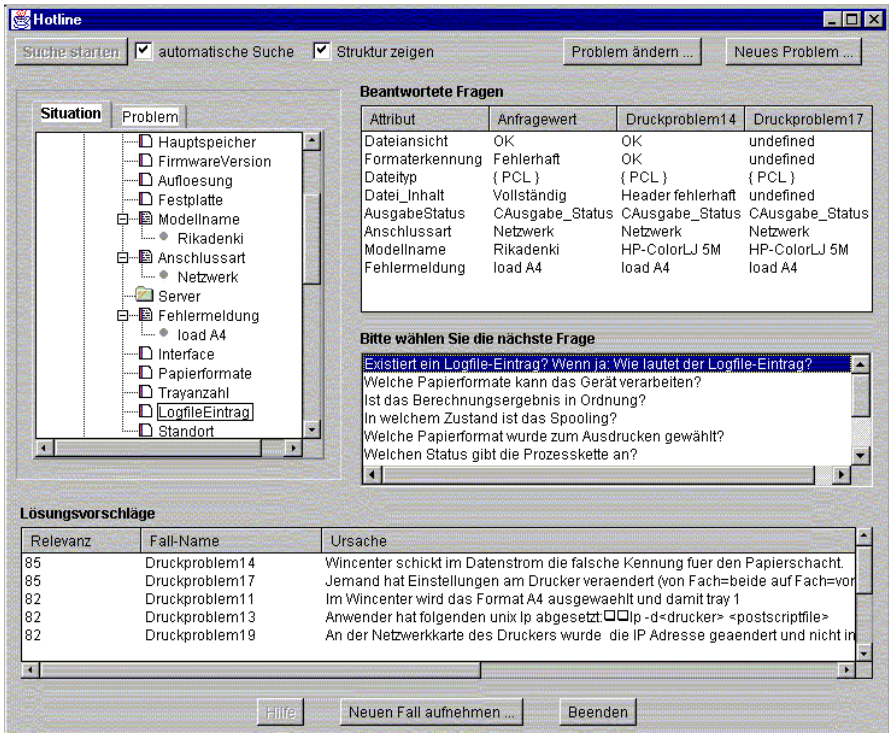

Figure 8: The Help-Desk Tool

contains the questions the system suggests to be asked to the user next.

The help-desk operator can either select the question to answer from the list or choose an attribute from the structure displayed to the left.

The list on the bottom of the screen shows the suggested solutions in their order of relevance. The help-desk operator can view these in the case-browser by double-clicking on them.

If the system was not able to find an appropriate solution in the case-base and the operator solved the problem in the conventional way, he/she can enter this solution into the system and store it along with the query (problem and situation description) and administrative information as a new case. The new case is stored in the case-buffer and later examined by the Case-Base Editor.

6 Lessons Learned

In the course of the project there were several aspects where the real world did not quite behave the way we had planned it.

First of all Knowledge Acquisition was (and is) much more time consuming than we expected. The main reasons for this are that the help-desk operators have little time to spare from their operational duties, are inexperienced in formulating their knowledge and that we want the resulting case-base to contain prototypical cases that are accurate and consistent. When we started the project we thought that we could utilise the 17000 trouble tickets that had been stored for the last years as cases. We soon found out that these lacked essential information and were useless (except for reminding the help-desk operators what else they had to take into account). Another surprising observation was that the resulting case-base contains less cases than we expected. We were able to cover most of the printer/plotter problems in the domain with less than 40 prototypical cases.

An aspect that we found to be essential in the development of the system is the design and layout of the user interface. Even though the help-desk operators are experienced computer users, they wanted the user interface to be clearly structured and not overloaded with information. They preferred to have less information on one screen to having everything on the screen at once and to switch to the respective screen whenever they needed structural or more detailed information.

The help-desk operators used the cases that were retrieved along with the suggested questions as guidance while solving the problem. Case focusing and question reordering based on information gain were perceived as very useful features in the help-desk system.

In situations where the system could not retrieve a solution, the help-desk operators used the object oriented domain model as a guide while solving the problem. The domain model also allowed us to make reliable similarity calculations and support the help-desk operator during case-entry. The structure of the domain also ensures that there are no ambiguities in the description of the cases and the problem definition.

The domain model allows us to use the same case-base in several languages. Since the model serves as a basis for the case-base, only the names of the classes, attributes

and values in the domain model have to be translated. The cases are instances of the domain model and stay the same. Since retrieval is based on the domain model, the retrieved result is independent of the display language. The translation effort is negligible compared to textual query/answer based systems where each case has to be translated and retrieval may result in different cases depending on the language.

7 Summary and Current Situation

Using INRECA technology, we were able to develop a system that stores the experience of the help-desk operators in a case-base and enables them to access, re-use and extend the knowledge in a natural and straightforward manner. The gains that we expect from this project are not only that we will ensure the availability of solutions to previous problems and that the knowledge of the help-desk operators will be cumulated and preserved, but also that the time needed to solve problems is going to decrease drastically. This will help to increase the productivity of the supported departments and the efficiency of the help desk.

Currently the first prototype is operational and under evaluation at the help-desk in Sindelfingen. Knowledge acquisition as well as the development of the system is continuing and we expect HOMER, the Hotline with Experience (HOtline Mit ERfahrung), to be fully operational at the end of 1998. The results of the evaluation so far are very encouraging both in terms of retrieval accuracy and user acceptance but have to be verified after knowledge acquisition as well as system development have been completed.

8 References

1. Thomas, H., Foil R., Dacus, J. : "New Technology Bliss and Pain in a Large Customer Service Center", in: Case-Based Reasoning Research and Development, Proceedings of the ICCBR97, Leake, Plaza (eds.), LNAI1266, Springer Verlag, Berlin, 1997, pp. 166-177
2. Althoff, K.D. et.al.: "Case-Based Reasoning for Decision Support and Diagnostic Problem Solving: The INRECA Approach", Proceedings, Fallbasiertes Schließen – Grundlagen und Anwendungen, 3rd German Case-Based Reasoning Workshop, B.Bartsch-Spörl, D. Janetzko, S. Wess eds., LSA-95-02, Fachbereich Informatik, Universität Kaiserslautern, 1995, pp 63-71

[i] Acknowledgement:

Funding for this work has partly been provided by the Commission of the European Union (INRECA-II: Information and Knowledge Reengineering for Reasoning from Cases; Esprit Contract no. 22196). The partners of INRECA-II are: Acknosoft (prime contractor, France), Daimler-Benz (Germany), tecInno (Germany), Irish Multimedia Systems (Ireland), and the University of Kaiserslautern (Germany). http://www.inreca.com

We would also like to thank the personnel of the CAD/CAM help-desk in Sindelfingen, especially Jürgen Fleisch and Bahram Abbasi, for their support.