*Workshop on Mixed-Initiative Case-Based Reasoning*, Fifth International Conference on Case-Based Reasoning, 24 June 2003, Trondheim, Norway. http://www.aic.nrl.navy.mil/~aha/iccbr03-micbrw/

# Adapting to the Level of Experience of the User in Mixed-Initiative Web Self-Service Applications

Mehmet H. Göker

Kaidara Software
330 Distel Circle, Suite 150
Los Altos, CA 94022

mgoker@kaidara.com

**Abstract.** Web self service applications deliver technical support to a diverse group of typically anonymous end users. Such applications have to be able to adapt themselves to the level of experience of the user and deliver consistent problems solutions. Since most users are not uniquely identifiable, creating a long term user model is not feasible. The knowledge in the application has to be extremely easy to create, update and maintain by a team of subject matter experts that are distributed world-wide and are not experienced in knowledge modeling. We describe an approach to build a user adaptive mixed initiative web self service application that can operate under these conditions and illustrate the ideas by analyzing an application.

## 1. Motivation

Web self-service applications provide customer support for technical equipment in a very cost-efficient manner. These applications have to be able to support a high number of users from different backgrounds and with varying levels of experience and consistently deliver correct solutions to problems. The user interface and the interaction modalities have to be able to support both novice and expert users. At the same time, the information in these systems and the user interface has to be easy to create, update and maintain.

Personalization techniques provide means to support users from different backgrounds and with varying preferences with a single application. Similar to human beings, a user adaptive or personalized system acquires and stores the characteristics or preferences of users in a user model and adapts the manner in which it acquires, selects and presents information accordingly (Langley (1997), c.f. Göker and Smyth (2001)).

Web self-service, troubleshooting and diagnosis applications are built to support users to find solutions to their problems as efficiently and effectively as possible. For such applications, the user's *personal* preferences regarding information display and interaction modalities play a secondary role. The primary goal is to understand what the failure is, determine the root cause of the failure, and find a solution to the existing technical problem as efficiently and effectively as possible. To achieve this goal, a troubleshooting system will need to obtain information needed to diagnose the problem from the user (and potentially other systems). Depending on the level of

experience of the user, the manner in which this information will be provided and can be acquired will vary and the system will have to adapt.

Adapting to the experience of the user is something done routinely for developing intelligent tutoring systems (c.f. Brusilowsky 1999). Some approaches used in tutoring systems require pre-processing or chunking the information into modules that can be sequenced or adapted according to the user's experience. This method can also be transferred to diagnosis problems (Brusilowsky and Cooper 1999). However, the overhead in creating and maintaining a complex knowledge representation poses a serious problem with respect to the long term maintainability of such an application. Especially in an application where cases are entered by authors and subject matter experts that are not experienced in knowledge modeling, the representation has to be extremely straightforward and unambiguous.

Alternatively some approaches adapt themselves to the user's level of experience by modifying the user interface, available interaction options and the presented information (e.g. Ardissono and Goy (2000); Fesenmaier, Ricci, Schaumlechner, Wöber, and Zanella (2003)). These approaches treat the experience of the user as part of a user model. However, building a user model requires the ability to uniquely identify a user. This is not necessarily possible for web self service applications.

We are interested in developing a web self-service application that allows users from different backgrounds to search for solutions to their technical problems. While some parts of the application will require user authentication, others will not. This prevents us from uniquely identifying users and creating reliable user models. The system is required to adapt itself to the level of experience of the user, acquire the necessary and sufficient information to diagnose a failure, and consistently retrieve the same solution for the same technical problem. The knowledge in the application has to be extremely easy to create, update and maintain by a team of subject matter experts that are distributed world-wide and are not experienced in knowledge modeling. This requires the representation to be very simple, easy to understand and unambiguous.

In the following, we describe and analyze the TAC Case collection[1] of Cisco systems as an application which adapts itself to the level of knowledge of the end user. The application provides network engineers support while troubleshooting Cisco equipment. Currently, the "wireless" section of the application is publicly available (http://www.cisco.com/public/support/wireless_launch.html) while other sections are password protected and only available to employees of Cisco customers. The analysis is based on the Kaidara Advisor[2] Software and the dialogue operators described for the Adaptive Place Advisor (Göker and Thompson (2000)).

## 2. Application Development with Kaidara Advisor

Kaidara Advisor is a development environment for building structural Case-Based Decision Support systems. Since it is a *structural* CBR development environment (Bergmann et.al (1999)), the resulting systems are based on an application specific

[1] Technical Assistance Center (TAC) Case Collection, © Cisco Systems, http://www.cisco.com

[2] © Kaidara Software, http://www.kaidara.com

domain model that contains the vocabulary (attributes, values, and their structure) for the application area. Cases and queries are a structured set of attribute-value pairs that are based on a representation derived from the domain model and stored as records in database tables. Kaidara Advisor provides the following three modes of interaction with the end user:

- ***Free Text:*** The end users can type in their query using free text. This query is parsed, ambiguities clarified, and used to create a structured query for the CBR Engine.
- ***Guided Search:*** Using information gain, Kaidara Advisor applications guide the end users by asking them to specify values for the most informative attributes, thereby guiding them towards a solution in an efficient and effective manner.
- ***Expert Search:*** For experienced users, Kaidara Advisor applications provide the ability to enter a query in a structured form that exposes part of the domain model. The user can also specify attribute weighting factors at query time.

Especially for troubleshooting and diagnosis applications of complex, technical equipment, we have found that novice users prefer to enter their problem description by using the free text entry modality or to be guided in a step-by step fashion. Expert users, on the other hand, may feel slowed down by the free text or guided modalities. Since they do not get intimidated when asked to characterize their problem by using the necessary and sufficient number of values in a structured form, they prefer the expert mode. However, even the most knowledgeable user will appreciate some guidance occasionally, and every novice will gradually learn. By seamlessly merging and linking the three modalities described above, user adaptive mixed initiative systems can be built. At the same time, the users still have the ability to use their preferred search method directly.

A typical interaction with a Kaidara system can start off with a natural language query. This query is parsed, ambiguities removed by means of clarification questions (which value is meant, which attribute is the value related to etc.) and an initial structured query send to the CBR engine. If the number of results returned by the retrieval engine exceeds a pre-set threshold, the system will automatically switch to guided mode, asking questions and guiding the user to a solution. If the users are not satisfied with the result set, they can modify the query directly by accessing the expert search mode.

## 3. User Adaptive, Mixed Initiative Web Self Service

As described above, we are interested in developing a web self-service application that is able to deliver consistent solutions to technical problems, independent of the level of experience of the user. The goal of such a system can be described as:

> *Acquire enough information to constrain the query in such a way that only a specified number of solutions above a given similarity threshold remain.*

To achieve this, the system has to adapt itself to the user's level of experience and how he/she chooses to provide information. Assuming that the interaction modalities described above are available, a typical TAC Case Collection session with a novice user would proceed as shown in Table 1.

**Table 1:** Sample interaction with a mixed-initiative diagnosis application (Novice)[3]

| | Interaction | Dialogue Operator | Description |
|---|---|---|---|
| *User* | It looks as if I am connected to the network but I cannot send any data. | PROVIDE_CONSTRAIN | Free text query |
| *System* | Do you mean your system is associated but cannot pass any traffic ? | CLARIFY | Parses query and asks clarification questions to remove ambiguities (Figure 1). Possible alternative attributes or values are also shown. |
| *User* | Yes | PROVIDE_CONSTRAIN | User provides constraint |
| *System* | Which products are you working with ? | ASK_CONSTRAIN | If there are no more ambiguities in the free text query, the system searches the Case-Base with the structured query. If the result set is too broad, the system switches to guided mode and asks the user to constrain his/her query further. |
| *User* | Access Point and a wireless card | PROVIDE_CONSTRAIN | User provides constraint |
| *System* | What is the state of your radio connection ? | ASK_CONSTRAIN | System tries to narrow down search further |
| *User* | Authenticated and Associated | PROVIDE_CONSTRAIN | User provides constraint (Figure 2) |
| *System* | Most similar cases are displayed in a table. | RECOMMEND_ITEM | System retrieves solutions from the case-base and displays these to the user (Figure 3). |
| *User* | Selects a solution from the result table | ACCEPT_ITEM | Views details of a case (Figure 4). |

The interaction is started by the user, but depending on the data provided, the system takes initiative to acquire more information. The user can modify the information he has provided before (*back*, *edit*), ignore the system's requests and provide responses to alternate questions (*skip*), ask the system to display what it has found so far (*submit*), or provide the answer to the current question and continue (*next*) (see Figures 1, 2). A user can also choose to switch modalities directly by selecting 'Free

[3] The example is taken from the public part of the Cisco TAC Case Collection (http://www.cisco.com/public/support/wireless_launch.html , © Cisco Systems, 2003). Due to the constant addition of cases as well as domain model, synonym and similarity metric upgrades, the retrieval results and interaction steps will vary over time.

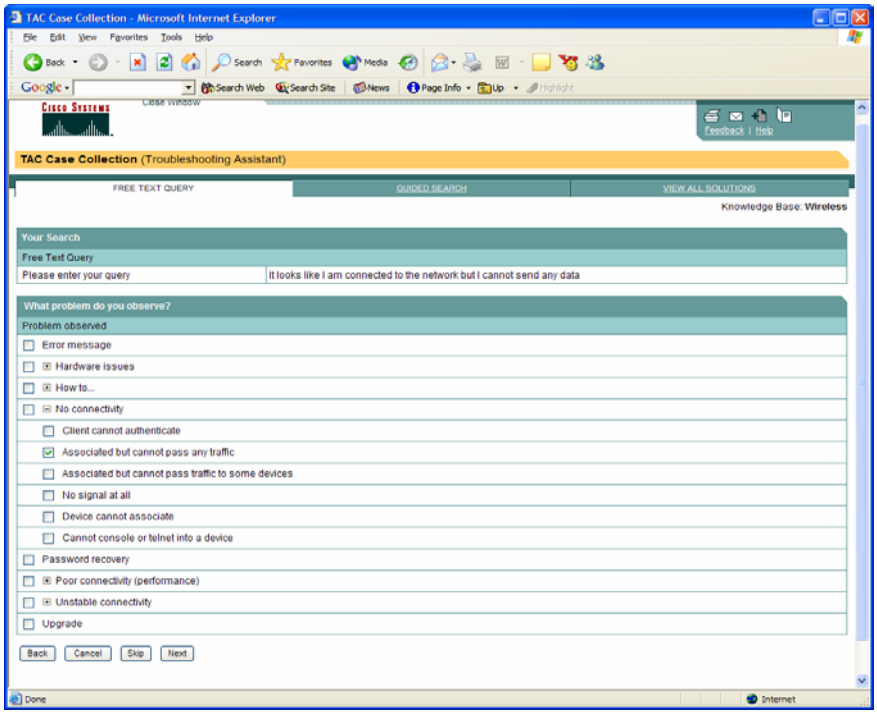


**Figure 1:** Clarification Question with the Cisco TAC Case Collection

text search', 'Guided Search', 'Advanced Search[4]' from the tabs on the top of the application. As such, the application is a mixed initiative system (c.f. Allen (1999), McSherry(2003)) where both sides can take control of the interaction when needed, where the user can provide volunteer information and recapture the control of the system at any time, and the system can select questions to guide the user.

The problem described by the user in Table 1 is caused by the client not being able to obtain a dynamic IP address from the DHCP server (Figure 4). The description of the novice is (as usual) rather symptom oriented and does not provide any relevant technical information. A novice also uses a terminology that differs from the one the expert (and the system) would use (Figure 1). This requires the system to parse and interpret the language of the user based on the synonyms and matching rules it has available.

Assuming the same symptoms and the same root cause, an interaction with an expert may look the way shown in Table 2.

[4] After analyzing the session logs of the application, 'Advanced Search' was removed from the user interface for standard users and replaced by 'View All Solutions". As expected, only very few users had been actually using the advanced search capability in this complex application domain..

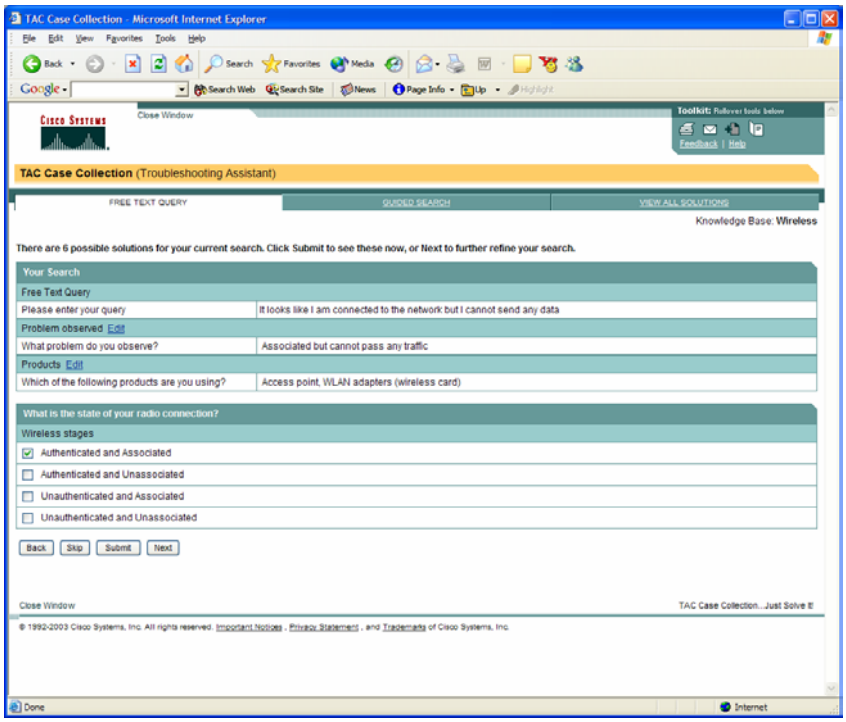


**Figure 2:** Guide Question with the Cisco TAC Case Collection

**Table 2:** Sample interaction with a mixed-initiative diagnosis application (Expert)

| | Interaction | Dialogue Operator | Description |
|---|---|---|---|
| *User* | Even though the client is associated and authenticated, it cannot pass traffic. The hardware is a AP 1100 and a patch antenna | PROVIDE_CONSTRAIN | Free text query |
| *System* | Problem Observed = *Associate but cannot pass traffic*;<br>Wireless Stage = *Authenticated and Associated*;<br>Product = *AP 1100*;<br>Antenna = *Patch Antenna*; | VERIFY_CONSTRAIN[5] | The system is verifying that the values it understood without interaction are correct (Figure 5). |
| *User* | Accept | ACCEPT_CONSTRAIN | User accepts constraint |
| *System* | Most similar cases are displayed in a table. | RECOMMEND_ITEM | System retrieves solutions from the case-base and displays these to the user. |
| *User* | Selects a solution from the result table | ACCEPT_ITEM | Views details of a case. |

While the problem and the result are identical, the interaction is different. The system extracts more (and different) information from the user's query, does not need

[5] VERIFY_CONSTRAIN and ACCEPT_CONSTRAIN are two new dialogue operators that were not present in the Adaptive Place Advisor. VERIFY_CONSTRAIN is a system operator that ensures that the system's interpretation of the user's natural language input is correct. ACCEPT_CONSTRAIN is the corresponding user operator to accept the system's interpretation.

to interpret as much, and, after verifying that it understood correctly, proceeds directly to the result display. Obviously, real life situations will be between these two extremes and novice users will learn the vocabulary of the system over time.

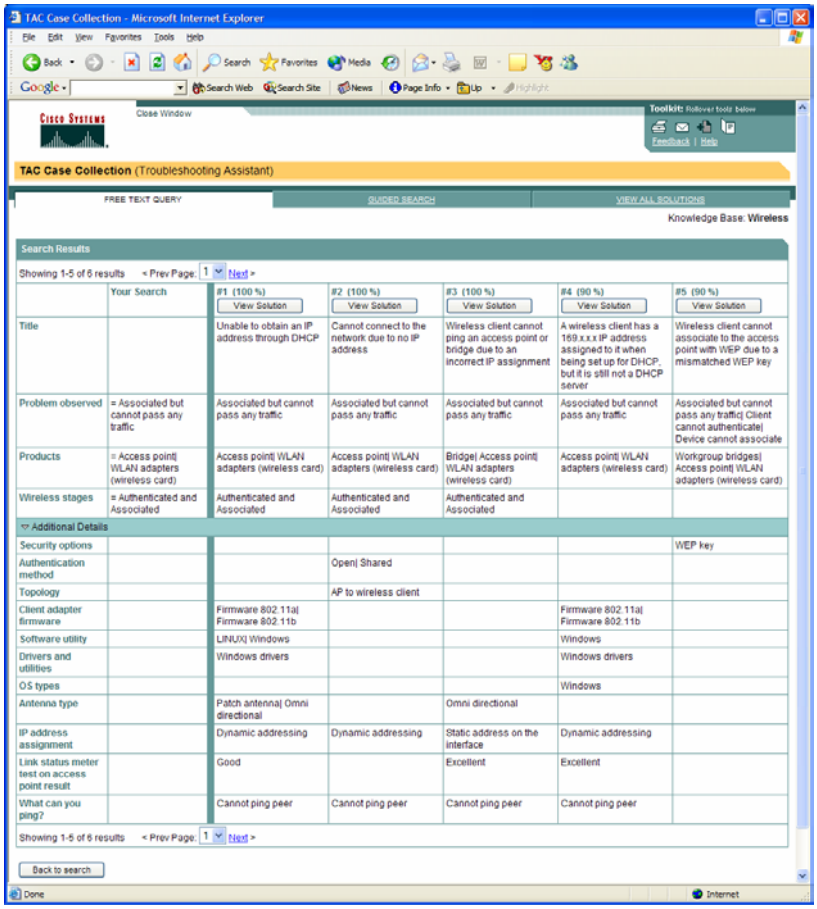


**Figure 3:** Result table of the Cisco TAC Case Collection

The TAC Case collection differs from other user adaptive applications in that it does not utilize a user model, only uses one simple knowledge representation scheme and only one user interface to adapt to varying levels of user experience. By using a mixture of natural language understanding, structural case-based reasoning, and information gain computations, the application ensures the user arrives at a solution in the fastest possible manner. In a troubleshooting environment, this will always be the goal and to the benefit of the users – independent of their level of experience or communication skills. The system can be thought of as a user adaptive system that is indifferent to the user. It does not really 'know' the level of experience of the user at any given time. Its only goal is to be able to obtain as much information as needed to determine what the root cause of the problem is. If a very inexperienced user copies an error message into the natural language text field and the system is able to identify

the problem and its solution immediately, it will have achieved its goal. And so will the novice who has a solution to his problem.

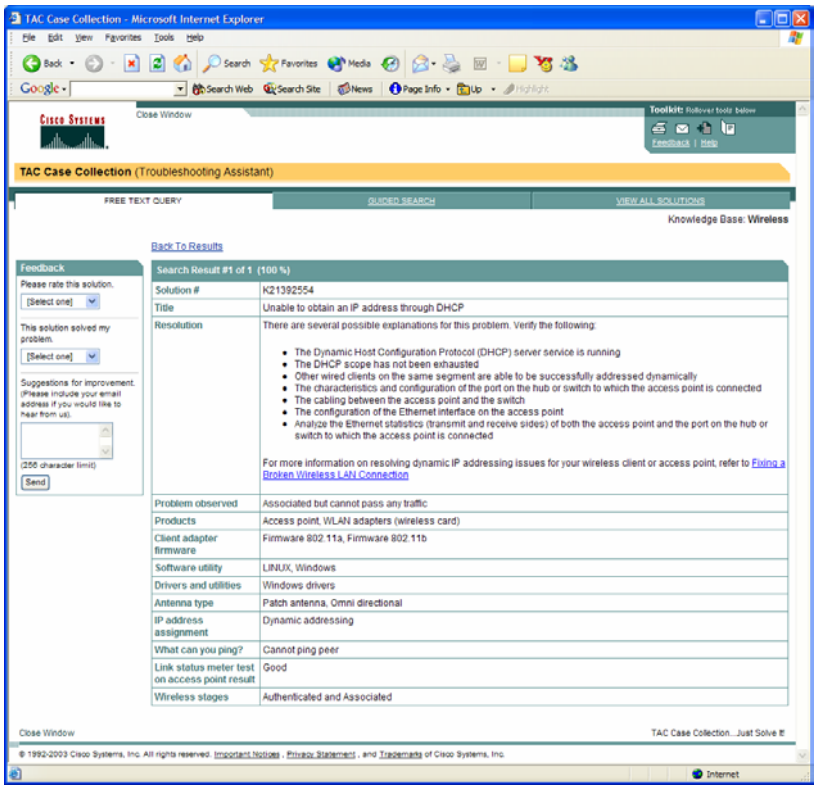


**Figure 4:** Case Details from the Cisco TAC Case Collection

The benefit of using a conversational, mixed initiative approach in terms of the ability to respond to user's needs independent of their domain experience has also been reported in Gupta, Aha and Sandhu (2002). The main difference between the two approaches stems from the representational differences of the underlying CBR systems. The application described here has been built using a structural CBR system that allows easy authoring and maintenance of cases by separating the domain model (with the associated hierarchical representation of attributes and values), the data (flat, individual records in a database table) and the interaction modality of the user with the application (generated by means of information gain computation for guided search). We believe that a representation methodology such as the one described in Gupta et.al. (2002) may be difficult to maintain given the complexity of the application domain, the high number of cases that are stored in the system and the number of concurrently working authors and subject matter experts.

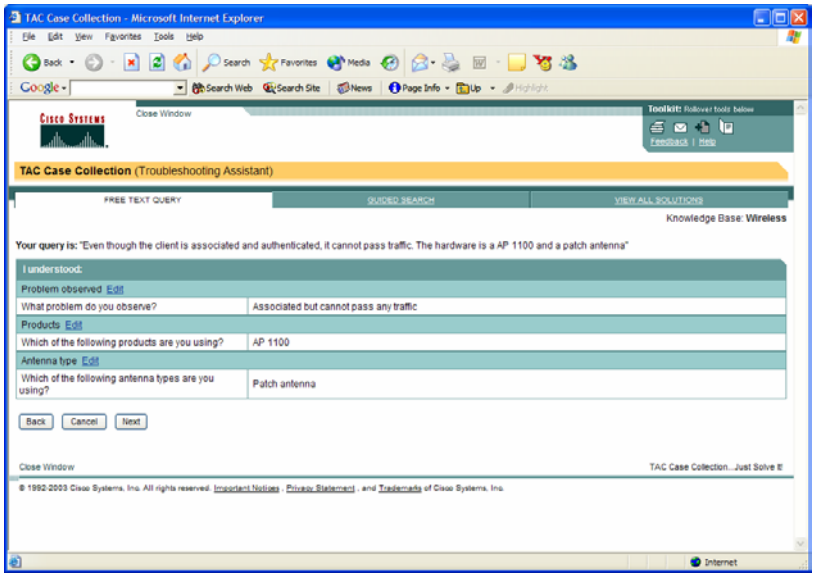


**Figure 5:** Information verification page of the Cisco TAC Case Collection

## 4. Summary and Future Work

We have described a mixed initiative diagnosis application that adapts itself to the level of experience of the user. While typical user-adaptive or personalized systems utilize a user model, modularized knowledge or configurable user interfaces to achieve this task, the described application makes use of a combination of natural language understanding and information gain based guided search to achieve this task. As such, it only uses one knowledge representation, a single user interface and no user model. By adapting the way it acquires information, it ensures that users arrive at a solution to their problem as fast as possible, independent of their level of experience. Since the web self-service environment cannot guarantee the availability of uniquely identifiable users, the system cannot create a long term user model. However, giving the users the ability to register themselves and store certain characteristics of their environment seems like a reasonable thing to do as a future step.

The application has been live on Cisco's web-site since April 2003. Currently we are collecting feedback and are planning to enhance the application accordingly.

**Acknowledgements:**

The author would like to thank David Aha, Pádraig Cunninham, Hector Muñoz-Avilla and Francesco Ricci for their extremely helpful comments and suggestions on the initial version of this paper.

## References


Allen, J. (1999), Mixed-Initiative Interaction. IEEE Intelligent Systems, Vol.6, pp. 14-16, 1999.

L. Ardissono and A. Goy. (2000), *"Tailoring the Interaction With Users in Web stores"* in "*User Modeling and User-Adapted Interaction*" vol. 10, N. 4, pp. 251-303. Kluwer Academic Publishers, 2000.

R. Bergmann, S. Breen, M. Göker, M. Manago, S. Wess, (1999), *"Developing Industrial Case Based Reasoning Applications: The INRECA Methodology"*, Lecture Notes in Artificial Intelligence LNAI 1612, Berlin, Springer Verlag, 1999

Brusilowsky, P.(1999), " Adaptive and Intelligent Technologies for Web-Based Education", In C. Rollinger and C. Peylo (eds.), Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, Gesellschaft für Informatik e.V. (GI), 1999, 4, 19-25,

Brusilovsky, P. and Cooper, D. W. (1999), *"ADAPTS: Adaptive hypermedia for a Web-based performance support system"* In: P. Brusilovsky and P. De Bra (eds.) *"Proceedings of Second Workshop on Adaptive Systems and User Modeling on WWW"* at 8th International Word Wide Web Conference and 7-th International Conference on User Modeling, Toronto and Banff, Canada, May 11 and June 23-24, 1999, Computer Science Report # 99-07, Eindhoven University of Technology.

D. R. Fesenmaier, F. Ricci, E. Schaumlechner, K. Wöber, and C. Zanella (2003), *"DIETORECS: Travel Advisory for Multiple Decision Styles"*, in Proceedings of Enter conference, Helsinki, Finland, January 29 - 31, 2003.

Göker M. and Thompson C. (2000), *'Personalized, Conversational Case-Based Recommendation'*, pp. 99-111, in E.Blanzieri and L. Portinale (eds.), "Advances in Case-Based Reasoning, Proceedings, 5th European Workshop on Case-Based Reasoning, Trento Italy, 6-9 September 2000", LNAI 1898, Berlin, Springer Verlag, 2000

Göker, M., Smyth, B. (2001), *"Delivering Personalized Information: What You Get Is What You Want"* in *"KI - Künstliche Intelligenz"*, Special issue on Artificial Intelligence and E-Commerce, R. Bergmann & Wolfgang Wilke (eds.), Gesellschaft für Informatik e.V. (GI), 2001, 01, 17-21

Gupta, K.M., Aha, D., Sandhu, N., *"Exploiting Taxonomic and Causual Relations in Conversational Case Retrieval"*, in " Craw, S. and Preece, A. (eds.), *"Advances in Case Based Reasoning, Proceedings of the 6th European Conference on Case Based Reasoning, ECCBR 2002, Aberdeen, Scotland, September 2002"*, LNAI 2416, pp. 133-147, Springer Verlag, Berlin, Heidelberg, 2002

Langley, P. (1997), *'Machine learning for adaptive user interfaces'*, in: 'KI97: Proceedings of the 21st German Annual Conference on Artificial Intelligence', pp. 53-62. Freiburg, Germany: Springer, 1997.

McSherry, D. (2002), *"Mixed-Initiative Dialogue in Case-Based Reasoning"*, in D. Aha (ed.), Proceedings of the Workshop on Mixed-Initiative Case-Based Reasoning at the Sixth European Conference on Case-Based Reasoning, ECCBR 2002, Aberdeen, 2002. http://www.aic.nrl.navy.mil/~aha/eccbr02-micbrw/