

KNOWLEDGE-BASED SALES ADVISORY: EXPERIENCES AND FUTURE DIRECTIONS

Dietmar Jannach and Markus Zanker

University Klagenfurt

Austria

Email: dietmar.jannach@uni-klu.ac.at, markus.zanker@uni-klu.ac.at

Keywords: e-Commerce and e-Business: B2C, e-Marketing, Business Solutions

Abstract: This paper summarizes our experiences gained from several industrial advisory applications that were developed with the knowledge-based ADVISOR SUITE framework over the last years and gives an outlook on future extensions of the presented system.

In the 'experiences' section of the paper, we first address aspects related to the development of such applications, such as knowledge engineering, software maintenance, or testing. In addition, we describe the main requirements for such an advisory application to be perceived as an intelligent, value-adding service by the end users and finally summarize the results of an industrial study on how advisory applications are able to influence the buying behavior of online shoppers.

The second part of the paper discusses current and future extensions of our system. The main lines of research addressed in this section are 'Extended debugging support', 'Automated extraction of product data from web sources', 'Log mining and advanced data analysis', and 'Community-adapted advisory systems'.

1 INTRODUCTION

Recommender systems are one of the most visible applications of Intelligent Systems and Artificial Intelligence technology. Today, the most prominent product recommendation systems are based on the analysis of the buying behavior of customers or on product ratings of a broad user community like on Amazon's online store. However, despite the broad success of collaborative or social filtering approaches, they are based on some particular requirements that limit their applicability to certain application types: First, they require that the user community has a significant size, such that there exists a sufficient number of ratings for the products in the catalog. In addition, these systems need some *ramp-up* time for new users for which no buying history is available. Finally, these systems do not work well for technical goods like digital cameras or TV sets, since for such domains the specific technical requirements of the customers have to be elicited for generating adequate product proposals. Such shortcomings can be overcome with the help of additional domain knowledge: Over the last decade, several content-based, knowledge-based, or hybrid approaches to rec-

ommendation have been proposed, see for instance (Bridge, 2001; Burke, 2000; Burke, 2002). However, in contrast to self-adapting community-based approaches, the main challenge when exploiting such domain-specific recommendation knowledge lies in the additional costs that come with the acquisition, validation, and maintenance of product data and the domain-specific recommendation business rules.

In this paper we discuss the experiences we made in several industrial projects with the ADVISOR SUITE system (Jannach, 2004), a domain-independent and fully knowledge-based framework for the development of online advisory systems. In particular, this system allows us to implement a more comprehensive approach to product recommendation (*'advisory'*): In applications built with ADVISOR SUITE, the customer is for instance guided through a sales conversation in a personalized way and is provided with additional information depending on his/her requirements and background knowledge. In addition, the system is also capable of explaining the users why a specific item is proposed to them and what are its advantages or disadvantages in specific situations.

In our experience report, which forms the first part of the paper, we discuss both the core issues of knowl-

edge acquisition and -maintenance (i.e., the development costs associated with such systems), engineering aspects, as well as more subjective issues like end-user acceptance or customer feedback. This discussion directly leads us to some future directions that we see in the domain of knowledge-based advisory systems. The four perspectives that we discuss in the second part of this paper are mostly related with future knowledge acquisition strategies, namely log mining and advanced data analysis, community-based development of advisory applications, extended debugging support and automated product data extraction from web sources. Before discussing these experiences and future directions, we will give a short overview on the ADVISOR SUITE system in the next section.

2 SYSTEM OVERVIEW

Figure 1 depicts an architectural overview of the ADVISOR SUITE system which consists of two major components: First, the system comprises a set of graphical tools for modeling all required pieces of knowledge which includes the core recommendation and advisory business rules, the personalization strategy, and finally, the respective presentation logic, which is all stored in a shared, underlying repository. The presentation *style*, which has to be adapted for each installation of the system, is defined with the help of page *templates* that also contain dynamic HTML code and which are automatically assembled by the system at run-time. The main design principle for all of these tools is simplicity, such that the domain experts (or web developers) can model the advisory knowledge by themselves such that knowledge engineering costs are minimized.

At run-time, the *Advisor Engine* dynamically evaluates and interprets the knowledge in the repository depending on the current state of the sales conversation. The *Personalization Agent* manages the current customer sessions, handles the user interactions, and determines the personalized dialog flow. In addition, the run-time components log all user interactions and provide functionality to store and reload advisory sessions.

In order to be able to behave intelligently upon the various types of user interactions and in order to provide useful recommendations, ADVISOR SUITE implements different techniques and algorithms that have their roots in the field of Artificial Intelligence, e.g.,

Rule based personalization: Interactive advisory dialogs built with ADVISOR SUITE can be fully personalized based on a rule-based mechanism, compare e.g. (Ardissono et al., 2003): Personalization of the web application is thus possible on different levels defined

by (Kobsa et al., 2001), like the *style* of presentation, the degrees of freedom in navigation, the actual content as well as the interaction strategy, or the level of detail in explanations, see (Jannach and Kreutler, 2005).

Knowledge-based query relaxation: The selection of adequate product proposals is based on a *filter-based* mediation between customer requirements and the characteristics of the offered products. In case of unsatisfiable user requirements, ADVISOR SUITE uses a novel, conflict-directed *query relaxation* approach (Jannach and Liegl, 2006; McSherry, 2004; Godfrey, 1997) for determining those products that fulfil as many of the customer's requirements as possible within the tight time limits of interactive applications. *Utility-based ranking of results:* Once the set of suitable products is determined, ADVISOR SUITE computes a personalized ranking of these products based on *MAUT* - the Multi Attribute Utility Theory (von Winterfeldt and Edwards, 1986), an approach that allows us to take the current user's interests into account when computing the utility value for a certain product (Ardissono et al., 2003).

Technically, the system is implemented with state-of-the art web technology: Java is used as programming language and the shared repository is implemented on top of a relational database system. The dynamic web pages are built with *Java Server Pages* technology; customization of the layout is done with *Cascading Style Sheets (CSS)*. Further details about the implementation of the system can be found in (Felfernig and Kiener, 2005; Jannach, 2004).

In the following, we will summarize our experiences made and discuss success factors when building and deploying applications with ADVISOR SUITE.

3 EXPERIENCES

Development process / Knowledge engineering. Up to now, we have developed around two dozen advisory applications for different domains such as consumer electronics, tourism, financial services, and even for goods of 'quality-and-taste' like fine cigars or wine. In summary, the main factors that in our view influence the success and efficiency of the knowledge acquisition process are as follows: Existence of a structured process, early user involvement, adequate tool support, and the background/motivation of domain experts. In fact, these aspects must not be considered in an isolated manner as they also influence each other. In our experience, the most successful development strategy for building advisory applications is based on an *evolutionary, prototypical approach* that includes the involvement of 'key users' right from the beginning, which is a common development prac-

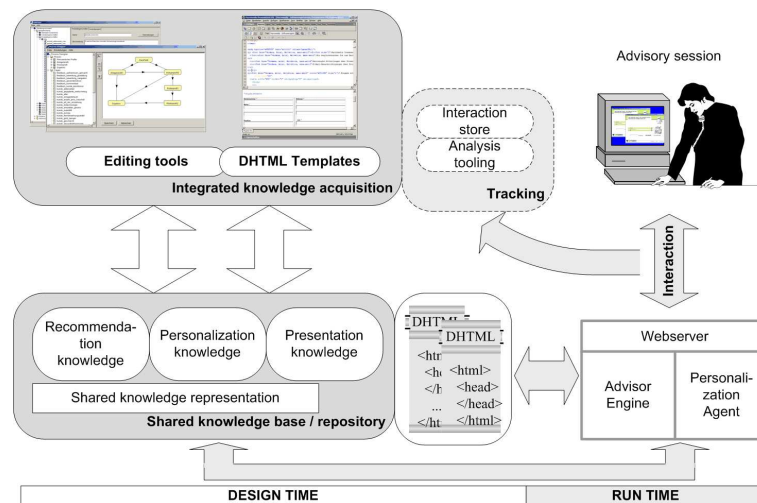


Figure 1: Overview of Advisor Suite architecture.

tice also for building other types of software systems. Our in-house process model thus includes initial analysis workshops in which the overall vision is elaborated and communicated between the stakeholders; having a particular eye on the *end user* is also crucial for the acceptance of the final system, since - as in many other projects - the stakeholders' expectations on the system's functionality may be different. For developing a common understanding of what the final system should be capable of, we extensively rely on rapid prototyping, i.e., we sometimes even use our software system during the workshops and collaboratively model and adapt the main pieces of knowledge (questions to be asked to the customer, recommendation rules). Our software tool provides means to automatically generate a working web application from this basic knowledge, such that the effects can be directly seen and further discussed.

There are two side effects, when key users can immediately see how their suggestions and knowledge are incorporated into the application: First, they experience that their ideas are being taken immediately into account and consequently their involvement in the project increases. On the other hand, these prototyping sessions also serve as a training on how to use the tool and learn what it is capable of, which is important as the domain experts and key users are typically not interested in studying manuals. Still, whether the domain experts will directly use the tool to manipulate the knowledge bases by themselves, depends on the skills, background, and interests of the individual expert. At least in some of the projects, the domain experts were using the tools by themselves, which suggests that the conceptual model used in the editing tools is simple and intuitive enough to be used

by non-IT experts: In particular the usage of a user-oriented terminology, graphical tools for 'modeling' the dialog flow, or an *if-then-style* representation of business rules were crucial for the acceptance of the tools. In other projects, the knowledge bases were further developed in subsequent design workshops. There, the knowledge engineer together with the domain expert formalizes the knowledge. For modeling non-standard behavior and introducing complex business rules into the system, software development skills are still being required; however, we see that most parts of the domain knowledge are covered by standard functionalities of the tool. Finally, quite interestingly, we also see that the size of the knowledge bases always remained at a still manageable size, i.e., only a few dozen business rules are typically required to cover the whole recommendation knowledge.

Engineering and technology aspects. From an engineering perspective, developing a framework for building advisory applications in arbitrary domains that also supports personalized user interaction is a challenging task: The following major aspects have to be considered in the design of such a system. First, advisory applications do not stand alone, but are nearly always integrated into an existing web-site or online shop. As such, the use of state-of-the-art web technology (like Java, Java Server Pages or relational database systems and web servers) is important in order to simplify the integration process. In addition, as in any standard software system, all data structures have to be domain-independent and all algorithms must be steered by meta-data, while at the same time extensibility hooks have to be provided for plugging in domain-specific procedures.

More specific to advisory systems than these first

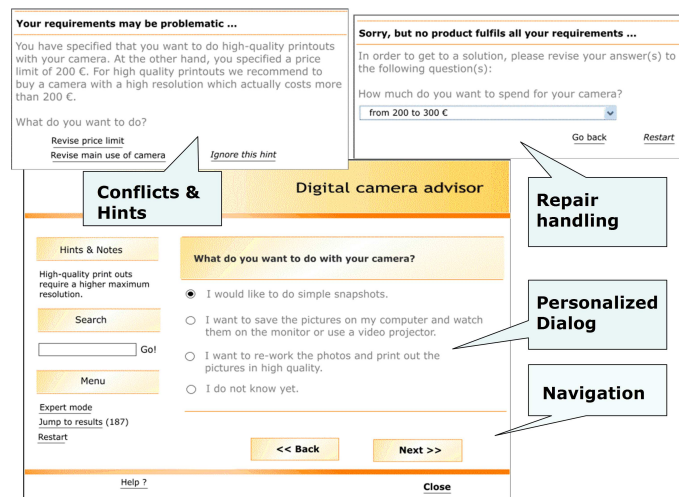


Figure 2: Screenshots of advisory application.

two aspects is the problem of minimizing and managing the interdependencies between the different layers of the application (data, logic, presentation). In particular, both the personalization and presentation logic builds upon the core definitions (e.g., which questions can be asked) and changes therein have to be immediately checked and/or reflected in the other layers. At design time, when the application is modeled, we address this problem by providing different views (and in fact different tools) on the knowledge, such that the individual tools do not become too complex. The most challenging engineering problem, however, lies in the development of the presentation layer which requires a very thorough design. One particular requirement of advisory applications is that the presentation style has to be easily adaptable by a web developer (for instance, because the layout has to be aligned with the corporate design of the online store), while the pages have to be highly dynamic such that changes in the knowledge bases (e.g., a new question) are immediately reflected in the application. These requirements are addressed in ADVISOR SUITE with the help of a specific 'template' mechanism and the use of so-called *custom tags*¹: In our approach, the final pages are assembled from small page fragments (e.g., how to display a question) that only contain standard HTML code, style sheets, and the above-mentioned *custom tags*. From the web developer's view, these custom tags appear like ordinary tags in the HTML code but actually provide advisory-specific functionality (like displaying all defined answers to a question) and in addition hide all implementation details, like the communication with the the advisor engine, personalization of defaults, the page flow and so forth. In our projects, we made excellent experiences with this template-based

¹ see <http://java.sun.com/products/jsp/taglibraries>

approach, which is actually not mandatory in our applications. In fact, it helped us to significantly reduce development and maintenance costs, when we compare such semi-automatically generated applications with manually engineered user interfaces on which we relied in previous versions of our framework. For more details on the implementation, see, e.g., (Jannach, 2004).

Intelligent behavior matters. We installed an advisory system on Austria's largest e-Commerce site (with respect to unique clients) that included over 100.000 user sessions. There an evaluation of over 1.500 feedback forms reported that the success and acceptance of an application heavily depends on whether the users attribute 'intelligent behavior' to it or not. The most important features in that context were that a) the system is capable of explaining the proposal in detail, b) that alternative solutions are proposed when none of the products fulfills all of the customer's requirements, and c) that the preference and requirements elicitation dialog is lively and personalized.²

Covering the first two aspects falls into the core strengths of knowledge-based approaches in general. In ADVISOR SUITE, for instance, we use inference traces, explicit explanation knowledge as well as natural-language text fragments to compile user-understandable explanations. In addition, the system implements novel algorithms for requirements relaxation and -repair to handle those cases, in which none of the products matches all customer requirements.

The third aspect (personalization of the dialog) is covered in our system with the help of explicit per-

²Details on the study can be found in (Zanker et al., 2004); another empirical study on the consumer behavior in the interaction with advisory applications is summarized in (Felfernig and Gula, 2006).

sonalization knowledge. Finding out what the customer's needs are is not trivial, even in real-world advisory or sales conversations. Therefore, our applications aim at simulating the behavior of an experienced sales person and guide the customers through an interactive dialog, in which the system asks questions, provides choices, and displays additional hints, help, or add-on information when this seems appropriate (see Figure 2). All this personalized behavior aims both at increasing the 'buying experience' for the customer while at the same time increasing his/her confidence in the system. For instance, no questions are asked that the user may not understand, or the system immediately responds on user inputs in a personalized way. Modeling this personalization knowledge of course induces additional knowledge acquisition costs. Still, we see that incorporating only a few personalized hints (or an animated avatar) can already significantly improve the liveliness and thus acceptance of the application. Again, the use of adequate modeling tools and the support for semi-automated generation of dynamic web pages is crucial for keeping the development and maintenance costs for such applications low.

Finally, note that in our system we deliberately do not rely on natural-language interaction for preference elicitation purposes. Since on the one hand, in particular novice users many times do not know which questions to ask, while on the other hand the users may attribute more intelligence to such a system than is actually warranted.

Effects of advisory systems are measurable. One of the most important questions from a business perspective is about measuring the *effectiveness* of such a system. Typical questions in this context are: Are the system's proposals adequate, and do the customers thus perceive recommendation as an added value of the online store? Do users trust in a system's recommendation and finally, does it mean that a recommender system influences and persuades online shoppers to buy?

(Adomavicius and Tuzhilin, 2005) summarize the state-of-the-art and open problems of measuring recommender system effectiveness for community-based approaches: While aspects like 'accuracy' and 'coverage' can be rather easily measured with today's technology, they claim that the question of measuring 'usefulness' and 'quality' is not fully answered yet and that also more economics-oriented measures that capture aspects like 'Return on Investments' or 'Customer Lifetime Value' need to be developed in future research.

In the context of industrial projects with ADVISOR SUITE, we have up to now conducted two studies for evaluating certain aspects of the effectiveness and usefulness of such applications. The first study in the domain of digital cameras - which was already men-

tioned above - was based on an 'online' analysis approach (Hayes et al., 2002), i.e., store visitors were directly asked for feedback about qualitative aspects of the advisory service. The evaluation basically corroborated our hypothesis that the advisory service was perceived as an added value by the customers: In particular, our investigation showed that the original motive of customers visiting the price-comparison platform has shifted already a few weeks after introducing the advisory service, i.e., 'getting the best deal' loses in importance compared to other aspects like 'getting an overview on the market' or 'finding the right model' (Zanker et al., 2004).

In the second study (Zanker et al., 2006), the question was whether an advisory service can significantly influence the buying behavior of online shoppers. For that purpose we analyzed the sales figures of an online store for premium cigars over a time period of three years (2002-2004). The advisory service has been deployed in May 2003. The product assortment comprises around 115 cigars from 18 different manufacturers; assortment and prices were basically stable during the whole evaluation period. One of the outcomes was, that the customers' buying behavior has significantly changed after the introduction of the advisory service: Before introducing *Mortimer*, the virtual cigar advisor, customers ordered the prominent makes like *Cohiba* or *Montecristo*. Afterwards, however, models of not so well known makes like *Juan Lopez Petit Coronas* entered the top-ten list of the most often sold items. However, we were not able to clearly relate advisory dialogues and orders in the online shop, because a logon was not required for using the sales advisor. Furthermore, online users must not place the order during the same online session, but may come back later on. For analysis we therefore evaluated the correlation between recommendations of the system and actual sales. Figure 3 displays therefore the increase in sold items versus the number of recommendations by the virtual sales assistant that have been explicitly acknowledged by the user (i.e. explicit clickthrough). Although we cannot state that there is a strong correlation between recommendations and sales (below 0.4 for our example), it nevertheless becomes evident that the propositions of the virtual cigar advisor influenced online users and helped to boost sales for specific models. Thus, exactly those models that were proposed in specific situations became more popular, for instance when users identified themselves as novices without smoking experience, specific models like *Juan Lopez Petit Coronas* or *Cohiba Siglo III* were recommended due to their taste and smoking duration. Overall, our first studies and experiences show that qualitative and quantitative effects of providing an advisory service on the corporate web site are directly measurable and that such measurements are of utmost importance

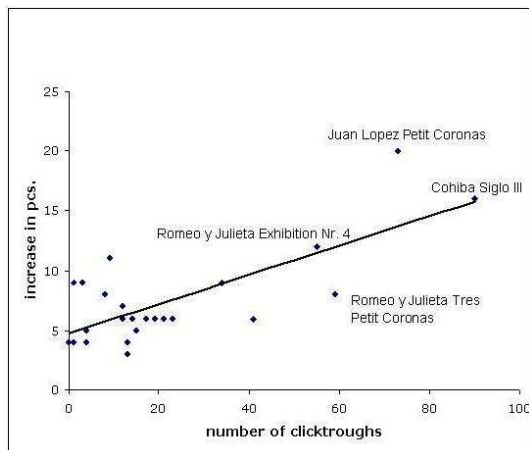


Figure 3: Correlation between clickthroughs and additional sold items

for the further spread of intelligent advisory applications, because many companies restrain from deploying such an online service as long as the potential Return on Investment is not clearly documented.

Finally - according to (Adomavicius and Tuzhilin, 2005) - we also see that more research in that direction is required and new techniques have to be developed for the process of assessing the real value of advisory and recommendation systems in industrial settings. Thus, one of our current extensions to the ADVISOR SUITE system is the development of a software framework for compiling various forms of statistics and for identifying patterns in the (change of) the consumer behavior in online stores.

4 FUTURE DIRECTIONS

Extended debugging support. Lack of adequate debugging tools is one of the major drawbacks of knowledge-based systems or expert systems in general. In our application domain, in which we aim at actively involving the domain expert in the development process, it seems even more important that the developer of the advisory application can for instance test complex recommendation rules, run regression tests upon changes, or let the system check the consistency of the definitions in the knowledge bases.

Our current work towards extended debugging support comprises the development two core components: The first one shall support the user in debugging manually engineered test cases, i.e., supporting the definition of cases and expected outcomes, storage and retrieval of cases, as well as automated regression testing, knowledge-base versioning and reporting. While the implementation of such a compo-

nent seems rather straightforward, the second component requires more intelligence as the development of new test and debugging approaches should comprise the following functionalities: First, it is important that the user gets adequate support in determining the 'good' test cases from the potentially vast set of possible ones. We therefore currently aim at automatically analyzing the possible interaction paths in the advisory application and generating representative test cases for different interaction patterns. In addition, we also try to exploit log data from past user interactions as they may help us to identify typical interaction patterns that were followed by a significant number of users.

Another area, in which more intelligent analysis tools can be helpful, is consistency checking. A typical problem, for instance, is to determine whether the recommendation rules are contradicting (and thus will never lead to a product proposal) or whether there are *dead ends* in the graph that represent possible user interactions. A first approach toward automated knowledge-base analysis for the latter example is described in (Felfernig and Shchekotykhin, 2006).

Automated product data extraction. In knowledge-based recommender systems, proposals are generated based on detailed knowledge about the items in the catalog. The quality of the proposal thus directly depends on the accurateness and completeness of the available product data. As our advisory applications typically run as an add-on to existing online stores or e-commerce platforms, parts of the product data are already available in electronic form. However, in many cases the quality of the existing data is not sufficient for building high-quality advisory applications on top of it: Either the data are incomplete or even incorrect, or they are not well-structured, i.e., only free-text or semi-structured descriptions are available. While manually maintaining product data is possible for small product catalogs that change infrequently, such an approach is in practice intractable in highly dynamic branches like consumer electronics.

In order to overcome these shortcomings, we have recently started a new funded project with a consortium of partners from academia and industry that aims at automatically extracting such product data from publicly available web sources like manufacturer home pages or other e-Commerce sites. Such an extraction process requires multiple steps like identification of relevant web pages, extraction of 'candidate' descriptions and key-value pairs, normalization, validation, and synthesization of contradicting or complementary data. Consequently, technologies from different fields like Information Retrieval, Information Extraction, Machine Learning, and Information Integration are required to accomplish the overall task. In our current approach, an explicit *domain model* (ontology) in the background serves as a starting point for

the extraction process. The domain model basically describes the structure of the data sets to be extracted, i.e., what specific characteristics should be identified for each product (e.g., the maximum resolution of a digital camera) and what the possible values for such an attribute are. Beside the domain model, also other forms of 'seed' knowledge like example data, extraction heuristics, and search patterns shall be exploited to improve the extraction results.

Due to the fact that also the domain model evolves over time, when for instance new features become available, we also aim at developing techniques such that the system also detects when the domain model itself could be extended, improved, or augmented: For instance, if a certain product feature can be found in many product fact sheets but it is not in the domain model, the system could make a suggestion to the domain engineer to extend the model accordingly.

Log mining and advanced data analysis. Advisory applications like those built with ADVISOR SUITE are highly interactive, i.e., the user continuously interacts with the system as (s)he specifies requirements, revises preferences, or browses and compares the items in the system's proposal. In our specific advisory framework, all user interactions are logged in the underlying database and are used for statistical and reporting purposes. However, we see that there is a lot of yet unexploited information and knowledge contained in these interaction logs and therefore, we currently aim at developing new techniques that shall help us to exploit this additional, implicit knowledge.

There are two different dimensions, in which we see a great potential for advanced data analysis: First, from the business perspective, the logs contain valuable information about the customers, in particular about their needs and preferences. In contrast to many other online surveys, users of advisory applications are interested in a high-quality personal recommendation, so we conjecture that they tend to answer the questions more thoroughly. A typical piece of information which can be useful for manufacturers could for instance be what features are really important for the customers, which are not, and for which (new) combinations of features there is a demand.

On the other hand, advanced log and data mining can help us to improve the advisory application itself. We can, for instance, locate critical conversation paths, i.e., situations when the advisory dialog is prematurely ended or questions are skipped because, e.g., this certain question is too complex for the users. In addition, the log data could also be used for continuously and automatically adapting the knowledge base itself. An example for this could be self-adjustment of priorities of the recommendation rules: In our applications, the system relaxes some of the recommendation rules in cases, when not all user requirements can be fulfilled. The relaxation is based on priori-

ties, i.e., an a-priori estimate on which requirements the users will be willing to compromise. With the help of the interaction logs, such estimates could be dynamically adapted such that they better match the customers' real intentions.

Community-adapted advisory systems. In our projects another experience was that users really appreciate additional sources of information with different viewpoints. Such add-on information includes glossaries, a discussion forum, product reviews, Frequently Asked Questions (FAQ), and so forth. While maintaining for instance such glossaries by hand is costly and time-intensive, we see more and more examples that such content can also be provided and updated by the user community at reduced costs. An example for such a project is the *Wikipedia*³ online lexicon, which is maintained by a broad user community and whose entries are of an astoundingly high quality. Of course, such a community-based approach is only feasible when the user community has a significant size, which means that it is suitable in advisory application for product domains with many potential (online) customers.

While installing adequate 'Wiki' software or setting up discussion groups is relatively easy and lots of tools are available, we currently investigate how we can go even a step further and involve the user community also in the process of improving the knowledge base of the advisory application. In particular in the domain of consumer electronics, we see from existing portals and e-commerce platforms that a lot of people are enthusiastic about sharing their experiences in a community or giving advice to other people in discussion fora.

In our first analysis we identified two basic complementary options how this community knowledge can be exploited in order to increase the quality and added-value of advisory applications. The first one is simply to link the existing pieces of information together. If, for example, a certain product is proposed to the user, (s)he can directly jump to the forum posts that are related with that product or view the glossary entries or FAQ for a certain technical feature.

The other, more complex option is to let (parts of) the users adapt, extend, or augment the contents of the knowledge base by themselves in the sense of a 'Wiki'. We think that building such a web-based maintenance and editing tool is not problematic from a technical perspective. The main challenge, however, is to build it in such a way that it will be usable for very heterogeneous groups of users; in fact, such a system and tooling has to be self-explaining as we cannot expect the users to read manuals. Although we made good experiences with our knowledge acquisition tools in industrial projects, in which it was

³www.wikipedia.org

possible to do some initial training, we conjecture that other conceptualizations and different editing alternatives will have to be provided for online users. Finally, a further major challenge in that context is the underlying problem of managing, integrating, validating, and harmonizing the potentially conflicting or inconsistent pieces of knowledge that are entered by the community.

5 RELATED WORK

Currently, recommendation systems built on Case Based Reasoning (CBR) technology form the most active sub-area of knowledge-based recommenders, for which (Lorenzi and Ricci, 2005) give a recent overview. The main topics in the area are e.g., query relaxation and query management, similarity measures, and comparison- or critique-based interactive critiquing, and hybrid systems (McGinty and Smyth, 2003; McSherry, 2004; Burke, 2002; Ricci et al., 2003). With regard to the user interface, recent work in the area also shows that rich multimedia presentations (Jiang et al., 2005) or personalized, conversational user interaction (C. A. Thompson and Langley, 2004) can help to improve the buying experience of the online shopper and increase the effectiveness of the overall system.

However, the authors are not aware of any recent research that addresses development and maintenance aspects of such knowledge-based systems, which, in our opinion, are crucial for the long-term success of such applications. In addition, research regarding to the user interaction is mainly focused on increasing the usability and end-user acceptance of the interface by e.g., adapting the interaction according to the current situation and user utterances. While these approaches are partially richer in their interaction style by supporting near-natural-language interaction (C. A. Thompson and Langley, 2004), our form-based approach, however, has the advantage that it is fully embedded into the development environment such that the strong interdependencies between presentation logic and recommendation logic can be stored in a central, comprehensive knowledge repository.

Research on the evaluation of highly interactive and knowledge-based recommender systems is still in its early stages; Most reported experiments in the domain of recommender systems perform an off-line analysis on an historical data-set (Herlocker et al., 2004). There, the predictive accuracy of algorithms is measured using historic log data. However, when a recommendation system is seen as an application that helps users to reduce information overload or even as a sales assistance tool more complex evaluation

scenarios are required. Missier and Ricci (Missier and Ricci, 2003) evaluated a travel recommender systems in an empirical study, where two versions of the systems encompassing different sets of functionality have been deployed. This way they could research the perceived usefulness of specific system functions and determine how the system influenced the information seeking behavior of users. A similar evaluation approach was taken by Felfernig and Gula (Felfernig and Gula, 2006): Their experiments showed that user satisfaction and trust of those users that received personalized product recommendations increased significantly versus those online-visitors that were solely allowed to browse through the product catalog alone. Furthermore, recommender systems that are intended to be sales applications that turn online visitors into buyers may not be reduced to their algorithmic properties alone. Appearance, usability as well as the situational context like time or expectations do influence their effects. Consequently, only a real-world setting where users actually spend their own money is appropriate to analyze the effects of recommendation technology on users' online-shopping behavior.

Therefore, we see our work as a further step in the direction of developing additional techniques of measuring the effectiveness of interactive online advisors and argue that a broad success of such intelligent e-services can only be reached if the potential Return on Investment for the merchants can be clearly demonstrated.

6 CONCLUSIONS

In this paper we have summarized the experiences of building intelligent advisory applications with the help of the fully knowledge-based and now commercialized ADVISOR SUITE framework. It has been demonstrated that such a comprehensive approach to product recommendation can serve as a valuable add-on service for the online customer and that the effects on the consumers' decision making and buying behavior can be directly measured. At the same time, our experiences show that the development and maintenance costs for such knowledge-intensive e-Business applications remain at a manageable level, when there is adequate, user-oriented tool support for the domain experts throughout the whole development process.

REFERENCES

- Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE*

- Transactions on Knowledge and Data Engineering*, 17(6):734–749.
- Ardissono, L., Felfernig, A., Friedrich, G., Goy, A., Jannach, D., Petrone, G., Schäfer, R., and Zanker, M. (2003). A framework for the development of personalized, distributed web-based configuration systems. *AI Magazine*, 24(3):93–108.
- Bridge, D. (2001). Product recommendation systems: A new direction. In Weber, R. and Wangenheim, C., editors, *Workshop Programme at 4th Intl. Conference on Case-Based Reasoning*, pages 79–86.
- Burke, R. (2000). Knowledge-based recommender systems. *Encyclopedia of Library & Information Systems*, 69(32).
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.
- C. A. Thompson, M. H. G. and Langley, P. (2004). A personalized system for conversational recommendations. *Journal of AI Research*, 21:393–428.
- Felfernig, A. and Gula, B. (2006). An empirical study on consumer behavior in the interaction with knowledge-based recommender applications. In *Proc. IEEE Joint Conference on E-Commerce Technology (CEC'06) and Enterprise Computing, E-Commerce and E-Services (EEE'06)*, San Francisco. IEEE Press.
- Felfernig, A. and Kiener, A. (2005). Knowledge-based interactive selling of financial services using FSAdvisor. In *17th Innovative Applications of Artificial Intelligence Conference (IAAI'05)*, pages 1475–1482. AAAI Press.
- Felfernig, A. and Shchekotykhin, K. (2006). Debugging user interface descriptions of knowledge-based recommender applications. In Paris, C., editor, *Proc. ACM International Conference on Intelligent User Interfaces*, pages 234–241, Sydney. ACM New York.
- Godfrey, P. (1997). Minimization in cooperative response to failing database queries. *International Journal of Cooperative Information Systems*, 6(2):95–149.
- Hayes, C., Massa, P., Avesani, P., and Cunningham, P. (2002). Online evaluation framework for recommender systems. In *Workshop on Personalization and Recommendation in E-Commerce*, Malaga.
- Herlocker, J., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22(1):5–53.
- Jannach, D. (2004). Advisor Suite - a knowledge-based sales advisory system. In *Proceedings of European Conference on Artificial Intelligence (ECAI/PAIS 2004)*, pages 720–724, Valencia, Spain. IOS Press.
- Jannach, D. and Kreutler, G. (2005). Personalized user preference elicitation for e-services. In Cheung, W. and Hsu, J., editors, *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce, and e-Service*, pages 604–611, HongKong. IEEE Computer Society.
- Jannach, D. and Liegl, J. (2006). Conflict-directed relaxation of constraints in content-based recommender systems. In *Proceedings of The 19th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA/AIE'06)*, Annecy, France. Springer. (to appear).
- Jiang, Z., Wang, W., and Benbasat, I. (2005). Multimedia-based interactive advising technology for online consumer decision support. *Commun. ACM*, 48(9):92–98.
- Kobsa, A., Koenemann, J., and Pohl, W. (2001). Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16(2):11–155.
- Lorenzi, F. and Ricci, F. (2005). Case-based recommender systems: a unifying view. In Mobasher, B. and Anand, S., editors, *Intelligent Techniques for Web Personalization*, pages 89–113. Springer Verlag.
- McGinty, L. and Smyth, B. (2003). On the role of diversity in conversational recommender systems. In Ashley, K. D. and Bridge, D. G., editors, *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 2003)*, pages 276–290, Trondheim, Norway. Springer.
- McSherry, D. (2004). Incremental relaxation of unsuccessful queries. In Funk, P. and González-Calero, P., editors, *Proceedings of the European Conference on Case-based Reasoning, LNCS 3155*, pages 331–345. Springer.
- Missier, F. D. and Ricci, F. (2003). Understanding recommender systems: Experimental evaluation challenges. In Weibelzahl, S. and Paramythis, A., editors, *Second Workshop on Empirical Evaluation of Adaptive Systems, held at 9th International Conference on User Modeling UM2003*, pages 31–40, Pittsburgh.
- Ricci, F., Venturini, A., Cavada, D., Mirzadeh, N., Blaas, D., and Nones, M. (2003). Product recommendation with interactive query management and twofold similarity. In Ashley, K. D. and Bridge, D. G., editors, *Proceedings of the 5th International Conference on Case-Based Reasoning (ICCBR 2003)*, pages 479–493, Trondheim, Norway. Springer.
- von Winterfeldt, D. and Edwards, W. (1986). *Decision Analysis and Behavioral Research*. Cambridge University Press, Cambridge, UK.
- Zanker, M., Bricman, M., Gordea, S., Jannach, D., and Jessenitschnig, M. (2006). Persuasive online-selling in quality & taste domains. In *Proc. 7th International Conference on Electronic Commerce and Web Technologies (EC-Web)*, Krakow, Poland. Springer.
- Zanker, M., Cech, J., and Russ, C. (2004). Geizhals.at: vom Preisvergleich zur E-Commerce Serviceplattform. In Salmen, S. and Gröschel, M., editors, *Handbuch Electronic Customer Care*, pages 295–306. Physica-Verlag, Heidelberg.