The VITA Financial Services Sales Support Environment

A. Felfernig[1], K. Isak[1], K. Szabo[2], P. Zachar[2]
[1]University Klagenfurt, Intelligent Systems and Business Informatics,
Universitaetsstrasse 65-67, A-9020 Klagenfurt, Austria
alexander.felfernig@uni-klu.ac.at, klaus.isak@ifit.uni-klu.ac.at
[2]Fundamenta Lakáskassza, Fundamenta-Lakáskassza Zrt., H-1922 Budapest
szabo.kalman@fundamenta.hu, zachar.peter@fundamenta.hu

Abstract

Knowledge-based recommender technologies support customers and sales representatives in the identification of appropriate products and services. These technologies are especially useful for complex and high involvement products such as cars, computers, or financial services. In this paper we present the VITA (Virtualis Tanacsado) financial services recommendation environment which has been deployed for the Fundamenta building and loan association in Hungary. On the basis of knowledge-based recommender technologies, VITA supports sales dialogs between Fundamenta sales representatives and customers interested in financial services (e.g., loans). VITA has been developed and is maintained on the basis of an environment which supports automated testing and debugging of knowledge bases and recommender process definitions. Besides presenting the VITA environment we focus on reporting empirical results which clearly show the payoffs of the deployed application in terms of time savings in the conduction of sales dialogs.

Problem Description

Complex and high-involvement products such as financial services impose increasing challenges on *sales representatives* responsible for identifying appropriate recommendations for customers as well as on *software developers* responsible for the development and maintenance of sales support environments. On the one hand, sales representatives have to know which services should be recommended in which context, and how those services should be explained. On the other hand, software engineering departments are facing the challenge of frequent change requests regarding service assortments and the rules specifying how those assortments are presented and offered to a customer.

Both aspects are major motivations for the application of knowledge-based recommender technologies (Burke 2000, Felfernig 2007) which allow a flexible development and maintenance of sales knowledge bases. Knowledge-based recommender technologies exploit deep knowledge about the product domain in order to derive recommendations. In contrast to the online selling of products such as books or movies (Herlocker et al. 2004, Pazzani 1999), the interactive selling of complex products requires the provision of intelligent technologies such as personalized dialogs, explanations, or repair actions for inconsistent customer requirements. An effective organizational embodiment of these technologies allows significant time savings for both sales representatives in the preparation and conduction of sales dialogs and software engineers in the development of the underlying sales support software.

From the business perspective, sales representatives must be effectively supported in the preparation, conduction, and completion of sales dialogs. The overall goal of financial service providers is to improve the performance of sales representatives in terms of an increasing number of sold products per year. Achieving significant time savings in all those phases is the door opener for a higher productivity of sales representatives since time savings can be exploited for contacting additional potential customers.

This situation was the background and major motivation for developing the VITA environment for the Fundamenta (www.fundamenta.hu) building and loan association in Hungary. The major goals of Fundamenta in the context of the corresponding project were the following:

- *Effective support for sales representatives* when conducting sales dialogs for Fundamenta products.
- *Significant time savings* in the preparation, conduction and completion of sales dialogs.
- *Improved sales performance* in terms of the number of sold products per year (as a direct result of the achieved time savings).
- Development of *intelligent sales support components* which allow an easy integration into the existing hardware/software environment.

The major reasons for integrating knowledge-based recommendation technologies in the VITA application were the following:

- *Reduced development efforts*: compared to conventional software development, development efforts can be significantly reduced using knowledge-based technologies (Felfernig 2007).
- *Effective maintenance*: testing & debugging support allows the reduction of knowledge acquisition bottlenecks. Thus a majority of maintenance tasks can be performed by domain experts autonomously.

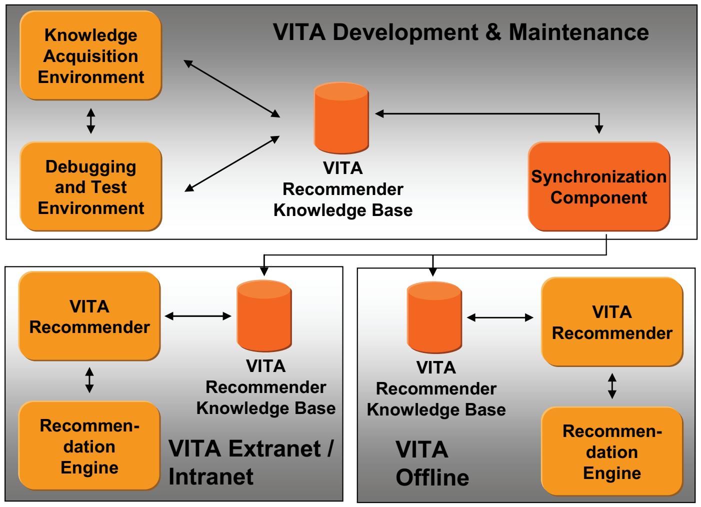

Figure 1: Overall VITA architecture.

- *Intelligent behavior*: for instance, the generation of explanations allows the automated derivation of advisory protocols which contributes to significant time savings in the order generation phase of a sales dialog (see Section *Application Use and Payoff*).

The remainder of this paper is organized as follows. Section *Application Description* provides an overview of the VITA sales support environment. The Section *Uses of AI Technology* discusses AI technologies integrated in the VITA environment. The Section *Related Work* provides a discussion of other existing approaches to the interactive selling of financial services. In *Application Use and Payoff* we analyse the impact VITA has on Fundamenta sales processes. The last two Sections deal with different aspects of VITA development and maintenance processes.

Application Description

VITA Recommender is a Web application supporting two basic scenarios (see Figure 1). First, VITA can be activated from a centralized server installation (VITA *Extranet/Intranet* on Win2003). This installation is used by Fundamenta sales representatives as well as by external agents independently offering financial services stemming from different companies. Second, VITA is installed as a *local application* (*VITA Offline*) on laptops of Fundamenta sales representatives (WindowsXP). Both are JSP-based Web applications automatically generated from the definitions in the VITA *Recommender Knowledge Base* (Hypersonic SQL database). Recommendation processes, product information and constraints are defined using the *Knowledge Acquisition* environment. Changes in the knowledge base are automatically *synchronized* with the VITA recommender applications.

In both scenarios a centralized *knowledge acquisition and maintenance* environment supports the graphical design of recommender knowledge bases and an automated testing and debugging of those knowledge bases (Felfernig 2007). This environment is domain-independent and provides basic functionalities for the effective implementation of recommender applications, such as automated application generation which supports rapid prototyping development processes. The environment is based on Java Web Start technology, which provides the basis for deploying Java-based applications on a Web Server and executing them on a corresponding client.

The *VITA Recommender Knowledge Base* contains all definitions needed for the automated generation of a JSP-based recommender application, for instance:

- *Financial service descriptions and instances* represent the service assortment which can be recommended to a customer.
- *Constraints* define which products should be recommended in which context (*filter constraints* representing specific marketing and sales strategies) and which customer requirements are (in)compatible. A priority defines the order in which constraints have to be relaxed in situations where no solution exists.
- *Process definitions* are represented as basic form of state charts which specify the way the recommender application should interact with the customer (Felfernig and Shchekotykhin 2006). Such process definitions contain questions which are posed to customers in the context of a sales dialog (see, e.g., Figure 5).
- *Customer properties*: each customer registered in VITA has a customer profile which consists of personal information, previous recommender session data (articulated preferences), and purchased products (as well from other financial service providers).
- *Session data*: for analysis purposes (e.g., mining association rules, clustering, etc.), each recommen-

dation session can be logged. Session data can be exported via XML interface. Currently, our environment supports the export of session data to WEKA (Witte et al. 2001).

- *Utility functions*: if defined, a utility function can calculate an optimal order for products part of a recommendation result. Products that best fit a customer's wishes and needs should be displayed first. Furthermore, utility functions can be exploited for ordering explanations and repair actions for inconsistent customer requirements.
- *Test cases*: for the purpose of *regression testing*, a set of test cases is stored in the recommender knowledge base. In the case of changes, this test suite helps guaranteeing the validity of a recommender knowledge base. Test cases are exploited by the diagnosis component for identifying faulty constraints (faulty transition conditions) in recommender knowledge bases (in recommender process definitions).

The calculation of a product recommendation is based on the execution of a conjunctive query on the given set of product tables. Consequently, constraints defined in the knowledge acquisition environment are translated into a corresponding query which determines relevant items to be recommended to a customer. Basic AI technologies integrated in VITA and in the corresponding knowledge acquisition and maintenance component are presented in the Section *Uses of AI Technology*. Currently, VITA is primarily applied for the recommendation of loans. Figure 2 depicts an example recommendation process for loans (the corresponding user interface is shown in Figure 3).

In the first phase of the loan recommendation process (*requirements elicitation*), the user has to provide *personal information* (age, income, etc.), the *time frame* of the loan, the *purpose* of the loan, and the *amount of money* needed. Based on this information, the recommendation process is forwarded to the next phase (*creditworthiness check*). In this phase, the customer has to provide additional details related to his current financial situation and needed financial and physical securities. In the case that no (temporary) solution can be found by the recommender application, for instance, because of a too high amount of required money, the recommender calculates alternatives (e.g., maximum possible credit amount) which in the following allow the recommendation of a loan. Having successfully completed the creditworthiness check, the process is forwarded to the *product advisory & selection* phase. Different loan alternatives are proposed (also redemption alternatives taking into account combination products in the form of insurances). Finally, the customer can select an alternative and step forward to the phase *detailed calculation & result presentation* where monthly redemption rates of the loan are determined.

An interesting functionality provided by the recommender is the *generation of an advisory protocol* which explains the reasons for the product recommendation in detail. Protocol generation is a very important functionality which follows the regulations of the European Union relating to an improved documentation of advisory sessions. Those explanations are automatically generated by the recommender application which otherwise would have been compiled manually. Examples for such explanations are *the recommendation fits to the regulations related to the maximum age of a customer* or *only loans with the required runtime have been recommended.*

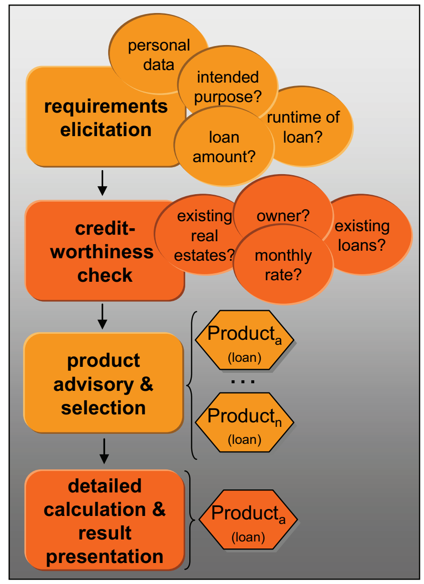

Figure 2: Loan recommendation process.

Uses of AI Technology

AI technologies used in VITA are documented in a variety of scientific publications, for instance see (Felfernig et al. 2004, Felfernig and Shchekotykhin 2006, Felfernig 2007).

The work of (Felfernig et al. 2004) provides a detailed overview on the application of Model-based Diagnosis (Reiter 1987) concepts to the identification of faulty constraints in configuration knowledge bases. This approach has been adapted (Felfernig 2007) and applied to the automated identification of faulty constraints in recommender knowledge bases. (Felfernig 2007) provides a detailed empirical analysis of the effects of applying Model-based Diagnosis concepts (Reiter 1987) for the identification of faulty constraints in recommender knowledge bases – the major results are significant time savings and error reductions in recommender knowledge base development and maintenance processes.

In (Felfernig and Shchekotykhin 2006) an approach to the automated identification of faulty transition conditions in recommender process definitions is presented. Such process definitions are basically finite state representations of the intended behavior of a recommender user interface (see, e.g., Figure 5). When defining such structures, faulty

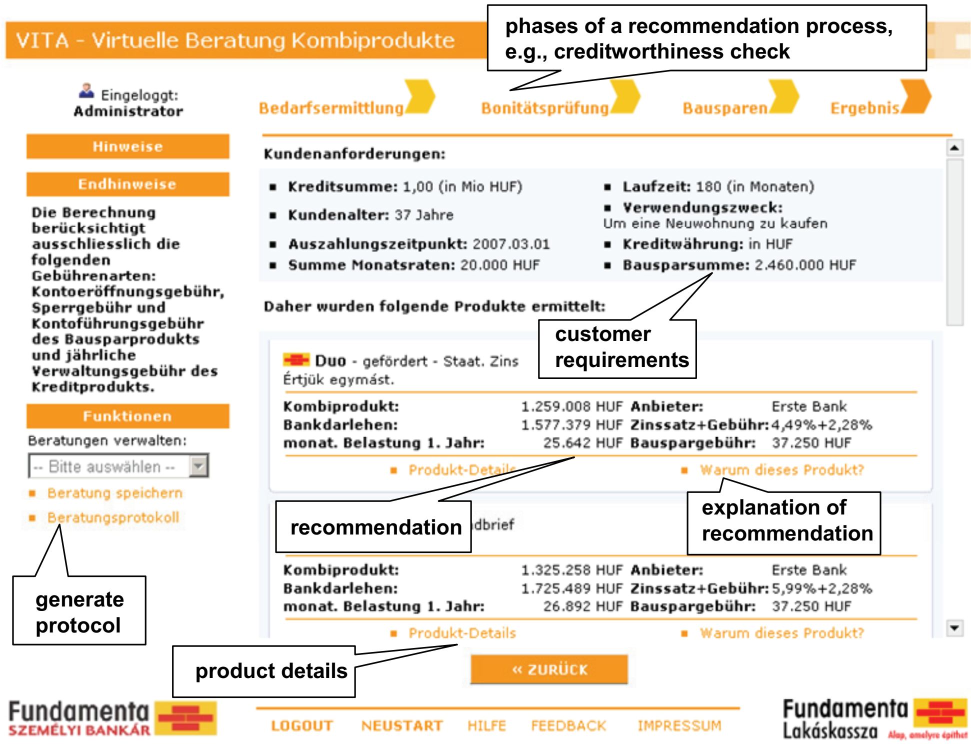

Figure 3: VITA user interface (result presentation).

transition conditions can be modeled which leads to inconsistent states in the interaction process.

Figure 4 provides an overview of technologies included in our recommender environment. We will now provide an overview on how those technologies are applied. Note that due to space limitations we had to restrict our examples to selected aspects, however, further examples can be found in other publications (see the references given above).

Multi-Attribute Utility Theory (MAUT) (Schmitt et al. 2003): this personalization approach is well known in knowledge-based recommender systems research. (Burke 2000) classifies this technology under the concept of utility-based recommendation which is a specific type of content-based recommendation.

We apply MAUT for the following purposes:

- *Ordering a recommendation set determined by the recommender application*. Each product part of a recommendation is evaluated w.r.t. to a given set of product dimensions. Conform to theories of cognitive psychology (Gershberg and Shimamura 1994), the most interesting products are presented first.
- *Ordering of repair actions*. In this case, a set of alternative repair actions is evaluated w.r.t. to a defined set of abstract product dimensions. Those repair actions with the highest probability of being accepted by the customer (highest utility) are presented first.
- *Ordering of explanations*. When presenting recommendation results (see, e.g., Figure 3), each of those results has an attached list of explanations (argumentations) why this result (product) fits to the wishes and needs of a customer. Depending on the given set of abstract product dimensions, explanations can be ordered (most important explanations are presented first). A simple example is depicted in Figure 6. Each filter constraint has an assigned explanation. Customer requirements such as *accessibility_of_investment* = *flexible* are evaluated w.r.t. their impact on different MAUT dimensions (e.g., dimension *Accessibility*). The higher the impact of certain requirements, the higher is the importance of the MAUT dimension for the customer and the higher is importance of the corresponding explanation. In our example, explanations related to the accessibility of a financial service would be displayed first. Note that in this simple example each customer requirement is evaluated w.r.t. to only one product dimension. In real-world settings, each requirement is evaluated w.r.t. a set of different dimensions.

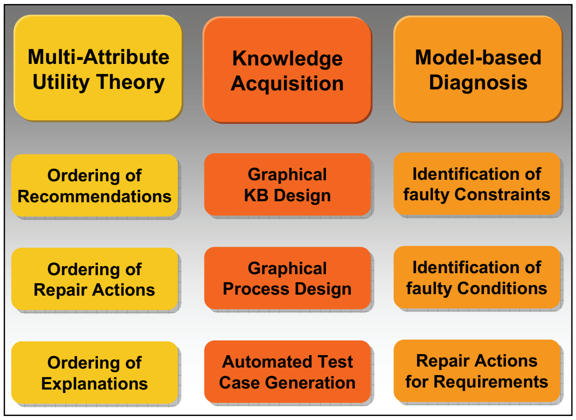

Figure 4: Used AI technologies in VITA.

Model-based Diagnosis (Reiter 1987): within the VITA environment, this approach is applied to the automated identification of

- *Inconsistent customer requirements*. If a given set of customer requirements does not allow the calculation of a recommendation, model-based diagnosis allows us to determine a *minimum* set of changes to the given set of requirements, s.t. the new set of requirements allows the calculation of at least one solution. Such minimal repairs allow keeping the set of requirements similar to the original set of requirements and thus can indicate interesting product alternatives for the customer.
- *Faulty constraints in recommender knowledge bases*. When changing existing recommender knowledge bases, in many cases faulty constraints are introduced. In this context, model-based diagnosis allows us to identify a minimal set of constraints which (if assumed to be faulty) help us to explain the faulty behavior of the knowledge base (wrong recommendations are calculated). The automated debugging of knowledge bases relies on test cases which specify examples for the intended behavior of a recommender knowledge base. Such test cases can automatically be generated by our test & debugging environment (Felfernig 2007).
- *Faulty transition conditions in recommender user interface descriptions*. Similar to the debugging of recommender knowledge bases, recommender process definitions (represented as finite state models) as well can be faulty. Such process definitions specify the intended behavior of a recommender user interface (Felfernig and Shchekotykhin 2006). Figure 5 depicts a simple example of a faulty user interface description. Consider the case where an expert (c_2) who is interested in financial advisory support (c_3) and long-term investments (c_6) is forwarded to the state *av*. Independent of the requirement specified for *av*, the dialog cannot be continued to one of the following states. *Well-formed process definitions* should not allow such situations. Therefore, we have developed concepts which support the identification of minimal sets of transition conditions which have to be changed in order to make the process definition consistent with a set of well-formedness rules. These concepts are not restricted to the application for knowledge-based recommender environments but can be generally applied to (single threaded) finite state models.

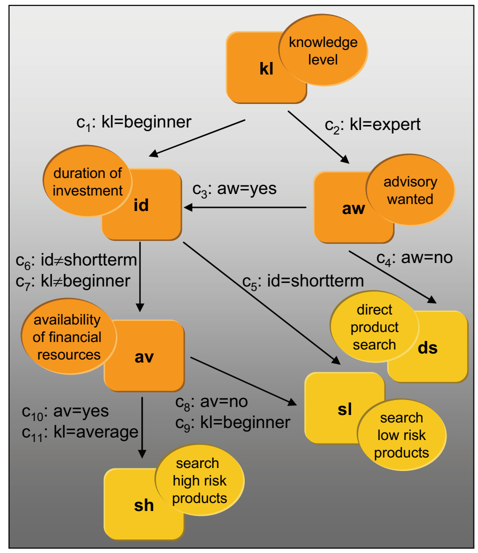

Figure 5: Faulty recommender process definition.

Empirical studies clearly show the applicability of our debugging approaches in terms of time savings when solving *error detection and repair tasks*. Results of an empirical study presented by (Felfernig 2007) report time savings of about 50% (!).

Knowledge Acquisition: Our knowledge acquisition environment supports the graphical design of recommender knowledge bases and process definitions. Due to space limitations, we omit a detailed presentation of this environment. An overview is given in (Felfernig 2007, Felfernig and Shchekotykhin 2006), where test case generation and debugging functionalities are discussed in more detail.

Consumer Buying Behavior: results of empirical studies (Felfernig and Gula 2006) clearly show the user acceptance of our recommendation technologies. However, different theories from the areas of decision psychology and cognitive psychology - see, e.g., (Gershberg and Shimamura 1994) - have to be analyzed in more detail in order to fully exploit the potential of knowledge-based recommender technologies. For example, primacy and recency effects indicate that the first and the last element of a list presented are more likely to be remembered by a person. This effect has different consequences on the application of recommender technologies (e.g., on the ordering of explanations, the ordering of questions, etc.).

Related Work

There exist a number of approaches to apply knowledge-based technologies in the financial services domain – see, e.g., (Stolze et al. 2000), (Leist and Winter 1994), (Felfernig and Kiener 2005) or (Tartakovski et al. 2005). (Stolze et al. 2000) show the application of knowledge-based configuration and personalization technologies for the selection of insurance products. The integration of feature-based search and product configuration is illustrated on the basis of application examples. Compared to our work, (Stolze et al. 2000) do not present concepts improving knowledge acquisition, testing, and debugging processes for recommender knowledge bases. Furthermore, no concepts are presented which support a utility-based ranking of explanations/repairs and the automated calculation of minimal changes to customer requirements in situations where no product could be found. (Leist and Winter 1994) motivate the application of knowledge-based configuration technologies for configuring insurance services. A meta-model for generic product structures is presented which takes into account structural properties of configurable products as well as additional restrictions related to the combination of different sub-components. Similarly, the meta-model of the VITA environment supports the representation of product properties and related constraints. Additionally, the VITA meta-model takes into account the design of recommender processes representing personalized navigation paths of a recommender user interface. (Felfernig and Kiener 2005) present an environment which is primarily applied for the interactive selling of *investment* products. From the application point of view, the major difference to the work presented in this paper is that the VITA environment is focussed on the interactive selling of *loans* which requires the provision of specific knowledge bases supporting creditworthiness checks and the calculation of pay back rates. In VITA, the recommender is the central component used for each loan-related sales dialog. In contrast, recommenders presented in (Felfernig and Kiener 2005) co-exist with the orginial CRM environment which as well allows product-centered sales dialogs. Both applications are based on knowledge-based recommender technologies where VITA is a further development of the environment presented in (Felfernig and Kiener 2005). From the technological point of view, the VITA environment supports additional debugging functionalities for recommender process definitions (Felfernig and Shchekotykhin 2006) which exploit complexity metrics for determining rankings for alternative diagnoses, additional ranking mechanisms for explanations and repair actions, and advanced test case reduction techniques.

As discussed in (Burke 2000), case-based reasoning (CBR) (Kolodner 1993, Vollrath et al. 1998) can be interpreted as specific type of knowledge-based recommendation approach. CBR is based on the concept of solving new problems on the basis of retrieving old problems likely to have similar solutions (Burke 2000). Case-based product retrieval searches in a database of product descriptions, where the retrieval of product recommendations is based on the evaluation of similarity metrics. In contrast to case-based recommendation, our approach implements a constraint-based product selection mechanism where filter conditions specify the set of products which should be selected from the offered assortment. An example for the application of CBR in the financial services domain (temporary life insurances) is presented in (Tartakovski et al. 2005). Temporary life insurances are contracts where insurers pay the insurance sum in the case of the death of the assured person within a specific period. In this period the customer has to pay a corresponding insurance premium. (Tartakovski et al. 2005) present an recommendation approach which determines life insurances on the basis of personal data and customer requirements. In this context, advanced concepts of CBR are taking into account generalized cases. A generalized case does not cover only a point of the case space, but a whole subspace of it.

Application Use and Payoff

The VITA financial services recommenders have been deployed in the 4th quarter of 2005 and are applied by about 800 sales representatives of the Fundamenta building and loan association primarily for the recommendation of loans. In 2006, about 125.000 products have been sold by the Fundamenta building and loan association. About 10% of those products have been sold on the basis of VITA. Since savings (about 90% of the sold products) are basic products without the need of an advisory support, no VITA processes have been implemented for this product type.

In order to demonstrate the payoff of the VITA application, we have interviewed sales representatives (n=205) of Fundamenta in Dec. 2006. The major goal of this study was to gain feedback on the current version of the recommender applications and to get an impression in which way the recommender application improves related sales processes. On an average, interviewees were working for Fundamenta since 3.0 years (std. dev. 2.8 years), the average age of the participants was 37.7 years (std.dev. 11.0 years). 38.6% were female participants and 61.4% were male participants.

The distribution between experts and beginners is shown in Table 1, where *experts* are selling all types of products from the Fundamenta product assortment, representatives with *average* expertise are selling 3-5 products, and *beginners* are focused on 1-2 types of products.

Expertise	beginner	average	expert
#	53	31	121
%	26	15	59

Table 1: Expertise of sales representatives.

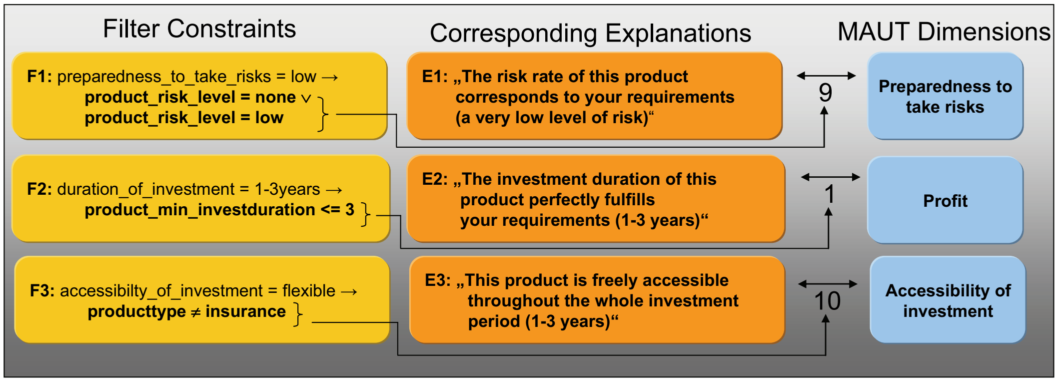

Figure 6: Simple example for the ranking of explanations.

Results of the study show that VITA is of high importance for improving business processes related to the selling of financial services (91.2% of the participants agreed on that aspect). On an average a sales dialog with the customer takes about 59.1 minutes (std. dev. 22.9 minutes) where the introductory part of an advisory session takes about 12.6 minutes (std. dev. 8.0 minutes), the concrete advisory time is about 34.1 minutes (std. dev. 15.8 minutes) and the completion of an advisory dialog (result presentation, order generation, etc.) takes about 16.2 minutes (std. dev. 7.9 minutes). The preparation of one customer visit takes about 15.7 minutes (std. dev. 16.6 minutes). Regarding the average duration of an advisory session, interviewees specified time savings directly related to the VITA application with about 9.0 minutes per advisory session (std. dev. 8.6 minutes), which means time savings of about 13.3% compared to the duration of advisory sessions without a corresponding VITA support. The significance of these time savings is confirmed by a corresponding t-test (t-score = 11.84, $p < 0.0001$). If we assume that an average sales representative conducts about 70 advisory sessions per year, this results in time savings of about 10.5 hours per representative per year. For high performers conducting about 500 sessions per year this means time savings of about 75.0 hours (!) per representative per year.

Due to the feedback of sales representatives, one of the major reasons for selling additional products were time savings achieved by the VITA application, where most of time savings have been achieved in the *detailed calculation & result presentation* (generation of advisory protocol) and the *creditworthiness check* phases. When comparing the sales rates of 2005 with the sales rates of 2006, the overall sales rate of VITA supported products has been increased by about 50% (!). On an average, sales representatives specified the increase of sold loans with 3.25 products per year (std. dev. 3.65 products). The significance of this increase is confirmed by a corresponding t-test (t-score = 9.59, $p < 0.0001$).

Assuming a low margin of 1% and an average loan amount of €30.000, the investment of about €100.000 for the implementation of the VITA environment has been amortized within the same year. The VITA application is of high importance for the reduction of errors in the offer generation phase (90.0% of the representatives agreed on that aspect). Furthermore, the majority of sales representatives (89.2%) articulated a high value of satisfaction with the recommender application. The reasons where time savings in the conduction of sales dialogs and more intuitive reporting and explanations for customers.

Application Development and Deployment

The investment for the development and deployment of the VITA system was about 1.5 man-years which includes the development of recommender knowledge bases, the translation of German recommender versions into Hungarian, the development and repeated adaptation of the graphical user interface, and the integration of the VITA recommender applications into the existing hardware and software environment. Knowledge acquisition, testing & debugging, and reasoning algorithms were already existing parts of the recommender development environment presented in (Felfernig 2007). On an average, three developers were involved in the VITA project. One of the major difficulties was the language barrier in the sense that VITA was our first application in a non German speaking country. Knowledge bases and process definitions have been initially developed in German and in the following been translated into Hungarian. We have identified *rapid prototyping* to be the best approach to an effective development a recommender application. Rapid prototyping follows the principle of concreteness where domain experts immediately see the effects of changing explanations, properties of products, images, transition conditions in process definitions and constraints in knowledge bases. On the basis of our knowledge acquisition environment (Felfernig 2007) such changes are performed on the graphical level – the corresponding models can be automatically translated into an executable recommender application. Such an automated application generation is extremely important for making the development of recommenders feasible for domain experts without a special IT education. VITA knowledge bases have been developed in cooperation with Fundamenta knowledge engineers having IT knowledge as well as knowledge about financial services.

Maintenance

Maintenance activities related to recommender knowledge bases as well as the VITA graphical user interface are conducted in cooperation with the Fundamenta IT department. Change requests are effectively processed on the basis of our graphical development and test environment (Felfernig 2007). 90% of the maintenance activities are related to changes in the underlying product assortment and constraints (new products, changing interest rates, etc.). Related change requests are collected and integrated into the knowledge base once a month. The remaining 10% of the maintenance activities are related to changes in the graphical interface and changes in the explanation and visualization of products. Such changes are systematically included in new versions released quarterly (10-15 change requests have to be processed).

Conclusions

In this paper we have presented the VITA environment which supports financial services sales dialogs. VITA provides significant time savings in the conduction of sales dialogs and thus acts as the basis for an increased productivity of sales representatives. VITA is a showcase for a successful application of knowledge-based recommender technologies in a commercial environment with a clear positive impact on existing business processes.

References

Burke, R. 2000. Knowledge-based Recommender Systems. *Encyclopedia of Library and Information Systems*, 69(32).

Felfernig, A.; Friedrich, G.; Jannach, D.; and Stumptner, M. 2004. Consistency-based diagnosis of configuration knowledge bases. *AI Journal*, 2(152): 213–234.

Felfernig, A. and Shchekotykhin, K. 2006. Debugging User Interface Descriptions of Knowledge-based Recommender Applications. Conference on Intelligent User Interfaces, 234-241, Sydney, Australia, ACM.

Felfernig, A. Reducing Development and Maintenance Efforts for Web-based Recommender Applications. *International Journal of Web Engineering and Technology*, 3(3):329-351, 2007.

Felfernig, A.; Kiener, A. 2005. Knowledge-based Interactive Selling of Financial Services using FSAdvisor, 17th Innovative Applications of Artificial Intelligence Conference (IAAI'05), 1475-1482, Pittsburgh, Pennsylvania, AAAI Press.

Felfernig, A.; Gula, B. 2006. An Empirical Study on Consumer Behavior in the Interaction with Knowledge based Recommender Applications, Conference on E-Commerce Technology (CEC06), 288-296, San Francisco, CA, IEEE.

Gershberg, F.; Shimamura, A. 1994. Serial position effects in implicit and explicit tests of memory. *Journal of Experimental Psychology*: *Learning, Memory, and Cognition*, 20:1370-1378.

Herlocker, J.; Konstan, A.; Terveen, L.; and Riedl, J. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22(1):5–53.

Kolodner, J. 1993. *Case-Based Reasoning*. M. Kaufmann.

Leist, S.; Winter, R. 1994. Konfiguration von Versicherungsleistungen. *Wirtschaftsinformatik*, 36(1):45–56.

Pazzani, M. 1999. A Framework for Collaborative, Content -Based and Demographic Filtering. *Artificial Intelligence Review*, 13(5-6):393–408.

Reiter, R. 1987. A theory of diagnosis from first principles. *AI Journal*, 23(1):57–95.

Schmitt, C.; Dengler, D.; Bauer, M. 2003. Multivariate Preference Models and Decision Making with the MAUT Machine, User Modeling 2003, 297-302.

Stolze, M.; Field, S.; and P. Kleijer. 2000. Combining Configuration and Evaluation Mechanisms to Support to Selection of Modular Insurance Products, 8th European Conference on Information Systems (ECIS2000), 858-865, Vienna, Austria.

Tartakovski, A.; Schaaf, M.; Bergmann, R. 2005. Retrieval and Configuration of Life Insurance Policies, ICCBR'05, 552-565.

Vollrath, I.; Wilke, W.; and Bergmann, R. 1998. Case-based reasoning support for online catalog sales, *IEEE Internet Computing* 2(4):47-54.

Witte, I.; Frank, E. 2001. *Data Mining*, Hanser.