Koba4MS: Knowledge-based Recommenders for Marketing and Sales

Alexander Felfernig

Computer Science and Manufacturing, University Klagenfurt,
Universitätsstrasse 65-67, A-9020 Klagenfurt, Austria,

Abstract. Due to the increasing size and complexity of products offered by online stores and electronic marketplaces, the identification of solutions fitting to the wishes and needs of a customer is a challenging task. Customers can differ greatly in their expertise and level of knowledge w.r.t. the product domain which requires sales assistance systems allowing personalized dialogs, explanations and repair proposals in the case of inconsistent requirements. In this context, knowledge-based recommenders allow a flexible mapping of product, marketing and sales knowledge to the formal representation of a knowledge base. This paper presents the knowledge-based recommender environment *Koba4MS* which assists customers and sales representatives in the identification of appropriate solutions. Based on application examples from the domain of financial services, basic Koba4MS technologies are presented which support the effective implementation of customer-oriented sales dialogs.

1 Introduction

Buying complex products (e.g. financial services, computers, etc.) is still a challenging task since many organizations offer simple query interfaces under the assumption that customers know the technical details of the offered set of products [1]. *Recommender technologies* [1,2,3,4,5,6,7,8,9] improve this situation by providing solution alternatives for the customer which are automatically derived from a set of customer requirements. There are three basic approaches to the implementation of recommender applications. *Collaborative Filtering* [4,6,7] is based on the concept of storing preferences of a large set of customers. Based on the assumption that human preferences are correlated, recommendations given to a customer are derived from preferences of a group of customers with similar interests, i.e. no deep knowledge about product properties is needed. Similarly, using *Content-based Filtering* [3,5], products are described by a set of keywords (categories) which are stored in a customer profile in the case that a customer buys a certain product. The next time, the customer enters the system, the stored preferences are used for identifying additional products which are assigned to similar categories. Additionally, there exist a number of approaches combining these basic approaches in order to gain an improved quality of the resulting solutions (see e.g. [10]). Finally, *Knowledge-based Recommender* applications (advisors) [1,2] exploit deep knowledge about the product domain in order to determine solutions exactly fitting to the wishes and needs of the customer. When selling complex products such as financial services, a

Felfernig A. (2005).
Koba4MS: Knowledge-based Recommenders for Marketing and Sales.
In *Proceedings of the 1st International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces*, pages 164-174
DOI: 10.5220/0001422601640174

customer's taste is not of primary concern - primarily solutions and explanations must be correct in every case (e.g., due to legal regulations), i.e., must correspond with the requirements articulated by the customer (see Section 3.1 for the repair of inconsistent customer requirements). This requirement can only be met by explicitly representing product, marketing, and sales knowledge [11,12], i.e. *Knowledge-based Recommender* applications (advisors) are the natural choice in this context.

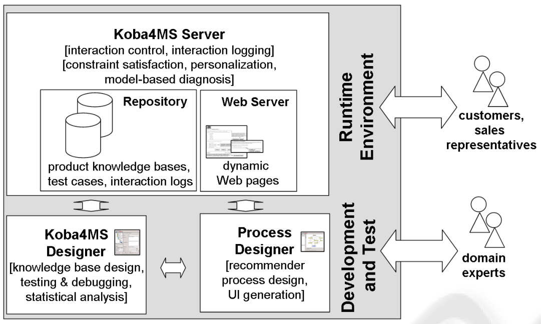

Koba4MS Server
[interaction control, interaction logging]
[constraint satisfaction, personalization, model-based diagnosis]
Repository
product knowledge bases, test cases, interaction logs
Web Server
dynamic Web pages
Runtime Environment
customers, sales representatives
Koba4MS Designer
[knowledge base design, testing & debugging, statistical analysis]
Process Designer
[recommender process design, UI generation]
Development and Test
domain experts

Fig. 1. Overall architecture.

In the following we give an overview of the major technologies implemented within the *Koba4MS*[1] environment, a domain-independent tool designed for the development of knowledge-based advisors. The major difference between *Koba4MS* and other knowledge–based recommender systems [2] is the inclusion of model-based diagnosis [13,14] and personalization techniques [1] which improve the effectiveness of advisor development as well as the interaction with the advisor. For example, a graphical development and test environment makes the implementation of advisors feasible for non-programmers, furthermore intelligent diagnosis and repair techniques actively support customers in situations where no solution could be found. *Koba4MS* can be applied in the following scenarios. Firstly, similar to traditional sales channels, improved sales assistance by *improving the accessibility of a product assortment on a companies Web page* generates added value for customers. On the one hand advisors allow an intuitive access to complex products for customers, on the other hand sales representatives are relieved from routine advisory jobs. Secondly, *sales representatives interact with advisors* when talking with the customer, where guided dialogs provide questions and explanations focusing on the customers wishes and needs. Throughout the paper we provide real-world examples from the financial services domain which is our leading application domain. Financial service advisory is a knowledge-intensive task which in many cases overwhelms customers as well as sales representatives. Therefore financial service providers ask for tools providing an intuitive access to their product assortment.[2]

The remainder of the paper is organized as follows. In Section 2 we present the *Koba4MS* recommender environment. In Section 3 we present examples for the usage of AI technologies which allow the implementation of knowledge-based advisors

[1] *Knowledge-based Advisors for Marketing and Sales* is a project funded by the Austrian Research Fund (agreement number FFF-808479).

[2] Products of financial service providers cover different areas of interest such as investment decisions, financing, pension, life insurance, etc.

(constraint satisfaction, model-based diagnosis and test, personalization, knowledge acquisition). Finally, Section 4 presents experiences from commercial advisor projects.

2 Koba4MS Environment

The *Koba4MS* environment (see Figure 1) provides comprehensive assistance for customers and sales representatives by supporting guided and personalized dialogs allowing an intuitive access to an assortment of complex products. In this context *Koba4MS* can be used for the following purposes.

- formalization of product, marketing and sales knowledge by non-programmers.
- testing/debugging knowledge bases in order to identify faulty constraint definitions.
- checking customer requirements for consistency and (in the case of inconsistencies) supporting a corresponding error handling.
- matching customer requirements to product properties, i.e. calculating a solution.
- diagnosing and repairing a set of inconsistent customer requirements, i.e. proposing minimal changes which allow the retrieval of a solution.
- explaining solutions in order to increase the confidence of the customer.

Koba4MS technologies are used in application domains such as financial services, digital cameras, cigars, computers, services in public administration etc. In the financial services domain *Koba4MS* technologies are applicable for the following reasons.

- Solutions must be objective, correct and explainable which makes approaches such as *Collaborative Filtering* or *Content-based Filtering* not the best choice.
- Typically, financial service providers want to develop advisors autonomously, i.e. knowledge representation formalisms are needed which allow the development of recommender knowledge bases for non-programmers (this is supported by graphical knowledge acquisition, model-based debugging and testing).
- Intelligent explanation, debugging, and repair mechanisms as well as automated test case generation are using model-based knowledge representations, i.e. deep knowledge about the application domain must be available (which is not available in *Collaborative Filtering* or *Content-based Filtering* approaches).
- Financial services recommendation is a complex task with a large number of constraints and possible solutions. In this context, knowledge-based approaches can significantly reduce efforts related to advisor development and maintenance.

Similar reasons motivate the application of knowledge-based advisors in other application domains such as online-selling of computers, digital cameras, etc.

2.1 Overall Architecture

Koba4MS product knowledge bases and process definitions are developed and maintained using a *Development and Test* environment (*Koba4MS Designer* and *Process Designer*). Products are defined within *Koba4MS Designer* or imported from external systems using an XML interface (for details see [1]). In the following advisors are automatically generated and made available for customers (e.g. online-stores, e-marketplaces, etc.) and sales representatives (e.g. intranet applications or installations on notebooks of sales representatives), where *Koba4MS Server* supports the execution of advisory sessions (*Runtime Environment*).

2.2 Development & Test Environment

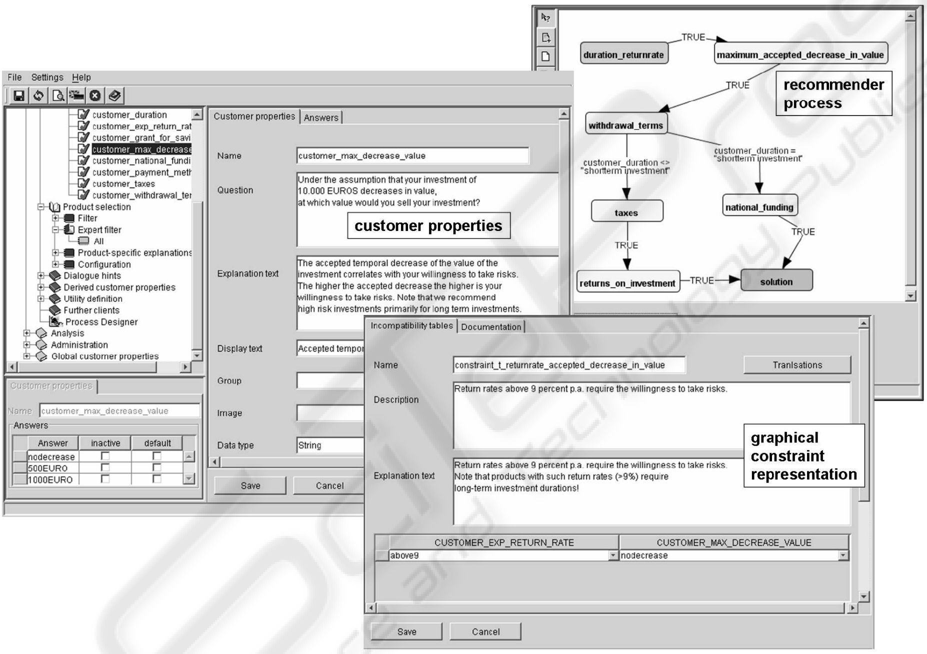
Koba4MS Designer. *Koba4MS Designer* is a graphical development environment for knowledge-based recommenders. It is based on Java Web Start which provides a browser-independent architecture for deploying Java-2 based applications on a client. The concepts implemented in *Koba4MS* are based on long-term AI research in the area of knowledge-based configuration and personalization [1,15,13,11,16]. *Koba4MS Designer* supports the design of advisors, where the relevant set of product- and customer properties is identified and transformed into a *recommender knowledge base* [11,12]. Such a knowledge base consists of the following parts (see Figure 2).

- *product properties* are structural descriptions of the provided products (e.g. life insurances can be characterised by the *possible length of life assurance policies*, *premiums of life assurance policies*, *links to additional product documentation*, etc.).
- *customer properties* are descriptions of possible customer requirements (e.g. within an investment advisory process the question *under the assumption that your investment of 10.000 EUROS decreases in value, at which value would you sell your investment*? is related to the *willingness to take risks*).
- *constraints* are restricting the combinations of customer requirements and product properties, e.g. *return rates above 9 percent require the willingness to take risks*. Constraints can be defined on the graphical level as well as on the textual level.

In order to support the analysis of advisors, *Koba4MS Designer* provides a statistical analysis component which operates on interaction logs of advisory sessions conducted by online customers or sales representatives.

Process Designer. A *recommender process* represents possible navigation paths which define the way the system adapts its dialog style to the knowledge level and interests of the customer. Such process definitions are based on a predicate augmented finite state recognizer (PFSR) [17] (constraints describe transitions between different states of a recommender process) which represents allowed navigation paths within an advisor (see Figure 2). Transition conditions between states of a recommender process are evaluated using the *Koba4MS* constraint engine. Based on a layout template definition, knowledge bases and process definitions can be automatically (no programming is needed) translated into an executable advisor (see e.g. Figure 3), where each state of the process definition corresponds to a Web-page in the generated application.

Testing & Debugging Knowledge Bases. The increasing size and complexity of recommender knowledge bases makes testing a critical task [18] in the context of successfully deploying and maintaining recommender applications. Process definitions (see e.g. Figure 2) are the basis for automatically generating test cases. Solutions (results calculated by the knowledge base) for generated test cases are presented to the domain expert who decides on their validity (*Result Validation*). Test cases deemed as correct by the domain expert are used for regression tests. Test case generation in *Koba4MS* follows a path-oriented approach (the test cases for each path are derived from the set of solutions to a corresponding constraint satisfaction problem) which allows a high degree of coverage [19]. The disposable time for testing is restricted, consequently mechanisms are provided which reduce the amount of tests without reducing the coverage of the overall test suite (except for random selections). Typically, domain experts agree with accepting efforts related to testing since solution quality is of serious concern. We can calculate a complete set of test cases which includes all possible transitions of a process definition, but this is only feasible for small and strongly constrained recommendation tasks. Approaches reducing the number of test cases are presented in [20].



Customer properties

Name: customer_max_decrease_value

Answers

Answer	inactive	default
nodecrease	□	□
500EURO	□	□
1000EURO	□	□

Name: customer_max_decrease_value

Question: Under the assumption that your investment of 10.000 EUROS decreases in value, at which value would you sell your investment?

Explanation text: The accepted temporal decrease of the value of the investment correlates with your willingness to take risks. The higher the accepted decrease the higher is your willingness to take risks. Note that we recommend high risk investments primarily for long term investments.

Display text: Accepted tempor

Group:

Image:

Data type: String

Name: constraint_t_returnrate_accepted_decrease_in_value

Description: Return rates above 9 percent p.a. require the willingness to take risks.

Explanation text: Return rates above 9 percent p.a. require the willingness to take risks. Note that products with such return rates (>9%) require long-term investment durations!

CUSTOMER_EXP_RETURN_RATE	CUSTOMER_MAX_DECREASE_VALUE
above9	nodecrease

Fig. 2. Definition of customer properties, constraints, recommender processes.

2.3 Runtime Environment

Koba4MS Server. The calculation of solutions for a recommendation task is based on constraint satisfaction problem solving [21]. Customer properties as well as product properties are represented as constraint variables. A solution for a given recommendation task (constraint satisfaction problem) is found if all constraints are satisfied. For an example screenshot of an interface see Figure 3. Note that Koba4MS allows the logging of advisory sessions which supports the improvement/fine-tuning of the actual knowledge base. Interaction logs can be analysed using the statistical analysis component integrated in *Koba4MS Designer*.

3 Used Technologies

Compared to *Knowledge-based Recommender* applications [1,2], *Collaborative Filtering* [4,7] and *Content-based Filtering* [3,5] do not exploit deep knowledge about the domain in order to determine solutions fitting to the wishes and needs of the customer. Using knowledge-based approaches, the relationship between customer requirements and financial services can be explicitly modelled in an underlying knowledge base [11]. Such model-based representations are the precondition for applying diagnosis and testing techniques.

3.1 Constraint Satisfaction

Search for Solutions. As already mentioned, *Koba4MS* problem solving is based on constraint satisfaction problem solving. A Constraint Satisfaction Problem (CSP) (C,V,D) [21] is defined by a set V of variables x_i, a set C of constraints c_j and a set D of domains d_i which defines for each variable the set of possible values. A CSP is solved if there exists a set of instantiations of the variables x_1, x_2, ..., x_n s.t. all constraints contained in C are satisfied. If no solution can be found by the search engine, constraints are *relaxed* starting with constraints with lowest priority. If nothing but non-relaxable constraints remain and no solution was found, a repair mechanism is activated. Solutions are presented in order of their usability for the customer (see Section 3, *Utility of Solutions*)[3]. In addition to constraints, *Koba4MS* supports *tips*, i.e. constraints representing e.g. *cross-selling opportunities* which are presented to the customer without interrupting the recommender process. Cross-selling conditions are integrated in the recommender knowledge as well represented as special types of constraints. A tip is: *long-term investments reduce risks, i.e. allow higher return rates than short-term investments without taking high risks*. An example for a tip representing a cross-selling rule is *a married sole wage earner with two children taking out a loan, is also a candidate for a risk insurance*.

Diagnosis and Repair of Requirements. If the result set is empty, conventional recommenders tell the user (customer) that no solution was found, i.e. no clear explanation for the reasons for such a situation is given. *Koba4MS* supports the calculation of repair actions for customer requirements (a minimal set of changes allowing the calculation of a solution). If Σ = {$x_1 = a_1$, $x_2 = a_2$, ..., $x_n = a_n$} is a set of customer requirements ($\Sigma \cup C$ has no solution), a repair is a minimal set of changes to Σ (resulting in Σ') s.t. Σ' $\cup$ C has a solution. The computation of repair actions [13] is based on the Hitting Set algorithm [14] which exploits minimal conflict sets (minimal sets $\Pi \subseteq \Sigma$ of variable instantiations triggering an inconsistency with C) provided by the constraint solver in order to determine minimal diagnoses and corresponding repair actions.

Test Case Generation. Automated test case generation (for more details see Section 2.2) is based on the definition of a constraint satisfaction problem. For this purpose a (complete) set of possible paths through a recommender process is determined. For each

[3] Critiquing of solutions [8] is within the scope of future versions of *Koba4MS*.

path a corresponding CSP is generated and executed - identified solutions represent test cases, i.e. possible settings of customer requirements.

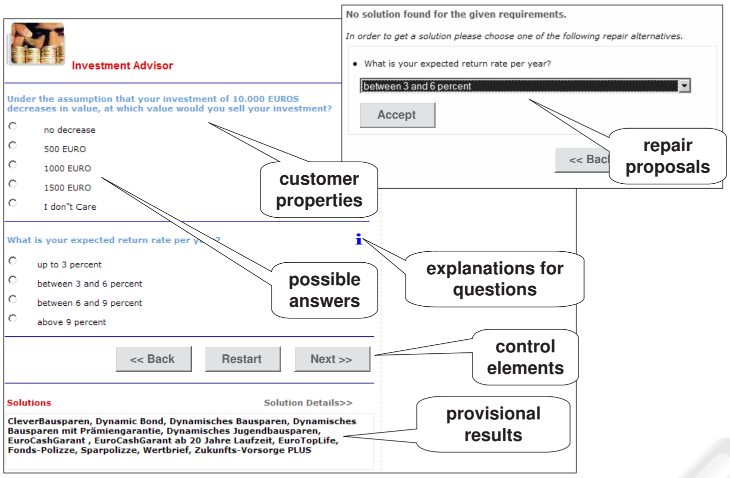
3.2 Personalization Concepts

Handling of Profiles. *Koba4MS* includes mechanisms allowing the adaptation of the dialog style to the user's skills and needs [1]. The user interface relies on the management of a user model that describes capabilities and preferences of individual customers. Some of these properties are directly provided by the user (e.g. *name or personal goals*, or self-estimates such as *knowledge about financial services*), other properties are derived using personalization rules and scoring mechanisms which relate user answers to abstract dimensions [1] such as *preparedness to take risks* or *interest in high profits* (dimensions describing the users interests) and *knowledge about funds, etc.* (dimensions describing the users knowledge about the domain). Values of the customer profile are collected from recommender sessions, missing values are asked the customer if they are needed in the context of a certain recommendation process (e.g. if the customer has not specified his/her age up to now then the related question is posed to the customer in the advisory dialog).

Dialog Style. Customers have different approaches to specify their requirements ranging from the direct specification of product parameters (e.g. a certain savings account running for 3 years) to a general specification of their personal goals (e.g. financing their children's education). An adaptation of the interaction style can significantly contribute to an improved approximation to the behavior of a human sales expert (an experienced sales assistant adapts his dialog style to the skill level and interests of customers). Depending on answers already provided by a customer, the dialog style can be personalized as follows.

- Alternative formulation of questions, e.g. questions posed to expert users can be differentiated from those posed to customers with less knowledge about the product domain.
- Rule-based formulation of default-answers, e.g. if the goal of the customer is to *put money by for a rainy day* the default answer to a question related to the maximum accepted decrease in value of the investment is *no value decrease accepted*.
- Alternative explanations for constraint violations, e.g. if the customer is a novice, a very general explanation about changes in the pension law is given, more detailed information can be included for experts.

Utility of Repair Proposals. If no solution can be found for a given set of customer requirements (although all relaxable constraints have been relaxed), *Koba4MS* provides a set of possible (minimal) repair actions which allow the calculation of a solution. Different customer properties have an assigned priority which indicates the importance of the variable for the customer. The *lower* the priority of the variable the higher the probability is that the variable is considered as focus of repair actions, e.g. if the type of returns on investment (at the end of the investment period, dividend payout) is not important



Investment Advisor

Under the assumption that your investment of 10.000 EUROS decreases in value, at which value would you sell your investment?

- ○ no decrease
- ○ 500 EURO
- ○ 1000 EURO
- ○ 1500 EURO
- ○ I don"t Care

What is your expected return rate per y ?

- ○ up to 3 percent
- ○ between 3 and 6 percent
- ○ between 6 and 9 percent
- ○ above 9 percent

Solutions

CleverBausparen, Dynamic Bond, Dynamisches Bausparen, Dynamisches Bausparen mit Prämiengarantie, Dynamisches Jugendbausparen, EuroCashGarant , EuroCashGarant ab 20 Jahre Laufzeit, EuroTopLife, Fonds-Polizze, Sparpolizze, Wertbrief, Zukunfts-Vorsorge PLUS

No solution found for the given requirements.

In order to get a solution please choose one of the following repair alternatives.

- What is your expected return rate per year?

between 3 and 6 percent

Fig. 3. Example user interface.

for a customer, this property is primarily considered as a potential candidate for repair actions. More formally, the personalization of repair proposals is based on the formula $f(x_1, x_2, ..., x_m) = \sum_{j=1}^{m} p(x_j)$, where $f(x_1, x_2, ..., x_m)$ represents the utility of repair actions related to the variables $x_1, x_2, ..., x_m$ and $p(x_j)$ denotes the customer-specific priority of variable x_j. Customer-specific priorities can be either defined statically or by a customer within the scope of an advisory session.

Utility of Solutions. A solution for a recommendation task is a set (portfolio) of financial services. The order of solutions should strictly correspond to the degree a solution contributes to the wishes of a customer. *Koba4MS* supports multi-attribute object rating [1], where each solution entry is evaluated w.r.t. to a predefined set of dimensions. *Profit*, *availability* and *risk* are examples for such abstract dimensions. Depending on the weighting of the dimensions for a specific customer (e.g. a customer is strongly interested in products with high return rates, i.e. compared to availability and risk, profit is a very important dimension) the set of solutions is ordered using the formula $g(x) = \sum_{i=1}^{n} e_i s_i(x)$, where n denotes the number of dimensions, $g(x)$ represents the utility of a solution x, e_i represents the interest of the customer in dimension i, and s_i is the contribution of solution x to dimension i.

Presentation of Solutions. For each solution a set of *immediate explanations* [16] is calculated, i.e. a set of explanations which are derived from variable assignments directly dependent on selections already made during search. Furthermore, *solution-specific explanations* are supported, e.g. if the customer is strongly interested in high return rates and a solution shows a remarkable return rate, this fact is explicitly mentioned when the solution is presented to the customer. In contrast to *immediate explanations* (derived in the search process), *solution-specific explanations* are related to explicitly defined explanation constraints.

3.3 Knowledge Acquisition

Knowledge Base Debugging. Effective debugging support for the implementation of recommender knowledge bases is a critical issue for a successful development and maintenance of advisors. In *Koba4MS* we have implemented model-based diagnosis algorithms [13,14] supporting the identification of minimal sources of inconsistencies in recommender knowledge bases. Similar to the diagnosis and repair of customer requirements, we apply model-based diagnosis techniques in order to identify a minimal set of constraints $\in C$ which - when deleted from the knowledge base - allow consistency restoration.

4 Experiences from Projects

A graphical development environment guaranteeing the maintainability of applications is a major prerequisite for successfully implementing a knowledge-based advisor. In the financial services domain the implementation and maintenance of knowledge bases must be supported for non-programmers, i.e. the knowledge acquisition component must provide intuitive modelling concepts. The overall efforts for implementing an advisor application are between one man month and about a man year where the most influencing factors are the complexity of the knowledge bases, the modeling knowledge of people engaged in the project and efforts related to the implementation of interfaces to remote systems (e.g. ERP or CRM systems). The correctness of solutions plays a vital role for the acceptance of the system by sales representatives applying the system while communicating with the customer. Domain experts do not use formal knowledge representation formalisms on a daily basis, i.e. effective *test and debugging* support is extremely useful and significantly improves the effectiveness of the overall advisor development process. Experiences from projects indicate a reduction of efforts related to knowledge base development of about 30-50 percent. The following conclusions can be drawn from the actual projects based on *Koba4MS* technologies.

- Knowledge Acquisition. Experiences from projects[4] show that graphical knowledge acquisition is a major precondition for enabling the design and maintenance of recommender knowledge bases and significantly reduces the knowledge acquisition bottleneck between domain experts and knowledge engineers. The usability has been shown, e.g., in our projects in the financial services domain, where domain experts autonomously develop and maintain the recommender knowledge bases after a first project where they were accompanied by a technical expert.
- Cross Selling. *Koba4MS* indicates cross-selling opportunities with a corresponding set of explanations as to why a solution is useful for the customer. The analysis of sales records e.g. in the digital camera domain shows significant improvements in the sales of add-on and niche products which were neglected previously.
- Routine advisory tasks. Effort reductions related to routine advisory tasks are reported, e.g. financial services advisory provided on the homepage relieves sales representatives from routine advisory jobs.

[4] See e.g. *www.hypo-alpe-adria.at* (investment advisor) or *www.geizhals.at* (digital camera advisor deployed on the largest Austrian online product platform).

- Documentation. Added value is provided by explanations for calculated service portfolios which are used as starting point for future advisory sessions. Furthermore, legal regulations can force companies to provide intelligent reporting for the customer, e.g. due to regulations of the European Union, financial service providers are forced to improve the documentation of advisory sessions - intelligent reporting is required which includes explanations as to why certain products were offered to the customer.
- *Koba4MS* knowledge bases are developed and tested by marketing and sales experts. Sales representatives can rely on the solutions calculated by the financial advisor and can provide qualified explanations.
- A set of applications has been implemented on the basis of the recommender technologies presented in this paper, e.g. the digital camera advisor PIXLA which was implemented for the largest Austrian online product platform (www.geizhals.at). This application exhibits about 10.000 successful advisory sessions per month. Users of www.geizhals.at were interviewed before and after the introduction of PIXLA. The major result of the study was a statistically significant increase of customer satisfaction (related to dimensions such as easiness to find products etc.).

5 Conclusions

In this paper we have presented the *Koba4MS* toolsuite which supports the implementation of knowledge-based recommender applications (advisors). The toolsuite is based on innovative AI technologies (model-based diagnosis, personalization, constraint satisfaction) which provide an intuitive access to complex products for customers as well as for sales representatives. *Koba4MS* includes a graphical development, test and debugging environment which allows the development and maintenance of recommender knowledge bases for non-programmers. The applicability of the presented concepts has been shown within the context of commercial projects. Next steps include the integration of critiquing mechansims into the presentation of results and the application of model-based diagnosis concepts to the identification of inconsistent transitions in recommender process definitions.

References

1. Ardissono, L., Felfernig, A., Friedrich, G., Jannach, D., Petrone, G., Schaefer, R., Zanker, M.: A Framework for the development of personalized, distributed web-based configuration systems. AI Magazine **24** (2003) 93–108
2. Burke, R.: Knowledge-based Recommender Systems. Encyclopedia of Library & Information Syst. **69** (2000)
3. Burke, R.: Hybrid Recommender Systems: Survery and Experiments. User Modeling and User-Adapted Interaction (2002) 331–370
4. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Trans. on Information Systems **22** (2004) 5–53
5. Pazzani, M.: A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Review **13** (1999) 393–408

6. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: ACM 1994 Conference on Computer Supported Cooperative Work. (1994) 175–186
7. Sarwar, B., Karypis, G., Konstan, J.A., Riedl, J.T.: Item-based collaborative filtering recommendation algorithms. In: 10^{th} Int. World Wide Web Conf. (2001) 285–295
8. Faltings, B., Pu, P., Torrens, M.: Designing Example-critiquing Interaction. In: International Conference on Intelligent User Interfaces (IUI'04), Madeira, Portugal (2004) 22–29
9. McSherry, D.: Explaining the Pros and Cons of Conclusions in CBR. In: ECCBR. (2004) 317–330
10. Shih, Y.Y., Liu, D.R.: Hybrid recommendation approaches: collaborative filtering via valuable content information. In: 38th Hawaii International Conference on System Sciences (HICSS'05), Big Island, Hawaii (2005) 217b
11. Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M., Zanker, M.: Configuration knowledge representations for Semantic Web applications. AI Engineering Design, Analysis and Manufacturing Journal **17** (2003) 31–50
12. Mittal, S., Frayman, F.: Towards a Generic Model of Configuration Tasks. In: Proceedings 11^{th} International Joint Conf. on AI, Detroit, MI (1989) 1395–1401
13. Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M.: Consistency-based Diagnosis of Configuration Knowledge Bases. AI Journal **2** (2004) 213–234
14. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence **23** (1987) 57–95
15. Felfernig, A., Friedrich, G., Jannach, D.: UML as domain specific language for the construction of knowledge-based configuration systems. IJSEKE **10** (2000) 449–469
16. Friedrich, G.: Elimination of Spurious Explanations. In: 16^{th} European Conference on Artificial Intelligence (ECAI 2004). (2004) 813–817
17. v.Noord, G., Gerdemann, D.: Finite State Transducers with Predicates and Identities. Grammars **4** (2001) 263–286
18. Preece, A., Talbot, S., Vignollet, L.: Evaluation of Verification Tools for Knowledge-Based Systems. International Journal of Human-Computer Studies **47** (1997) 629–658
19. Edvardson, J.: A Survery on Automatic Test Data Generation. In: 2^{nd} Conference on Computer Science and Engineering CCSSe'99. (1999)
20. Felfernig, A.: Koba4MS: Selling Complex Products and Services using Knowledge-based Recommender Systems. In: 7th International IEEE Conference on E-Commerce Technology, Munich, Germany (2005) to appear
21. Tsang, E.: Foundations of Constraint Satisfaction. Academic Press, London (1993)