TOOLKITS SUPPORTING OPEN INNOVATION IN E-GOVERNMENT

Alexander Felfernig, Christian Russ
Institut fuer Wirtschaftsinformatik und Anwendungssysteme, Universitaet Klagenfurt
Universitaetsstrasse 65-67, A-9020 Klagenfurt

Manfred Wundara
Informationstechnologie Magistrat Villach
Rathausplatz 2, A-9500 Villach

Keywords: e-government, open innovation, artificial intelligence, virtual advisors.

Abstract: Today there exists a variety of efforts to bring public administration closer to its customers (citizens, entrepreneurs, etc.). This paper investigates the concept of open innovation, i.e. the integration of the customer into the product creation process of a company, w.r.t. its applicability in the area of e-government. The concept of open innovation is well known within the context of mass customizing products and services, i.e. production and selling of customer-individual products and services under mass production pricing conditions. The authors show how approaches from the area of artificial intelligence can be applied as tools for open innovation in e-government.

1 INTRODUCTION

Henry Ford stated that his "customers can get every car color they want as long as it is black". This is a perfect characterization of the *mass production* paradigm. Ford was not confronted with any competitors selling red or blue cars worldwide via online shops. Customers did not ask for different equipment variants, they were satisfied with one basic type of car. However, times have changed and todays enterprises are forced to continuously place product innovations on the market. Within this context the paradigm of *mass customization* (Anderson, 1997; PineII et al., 1993) has been established propagating the production and sales of customer-individual products and services under mass production pricing conditions. However, from the customer point of view the mass customization paradigm per se does not guarantee that the customer gets the products or services he needs and he is looking for, it only guarantees a highly variable product sortiment with competitive development, production and distribution costs. Consequently, the paradigm of mass customization has to be enriched with the aspect of customer orientation, i.e. the integration of the customer into the product creation process of a company. The literature entitles this approach to customer integration as *open innovation* (Tseng and Piller, 2003; Piller and Stotko, 2003), where companies actively provide customers with means to articulate what they want and carefully listen to customer requirements in order to generate innovations that meet or maybe exceed the needs of their customers. Tools provided to the customer in this context are entitled as *tools for open innovation* in the literature (VonHippel, 2001). Such tools support user-centered dialogs by imposing personalized questions and provide explanations throughout the advisory process. Results of the advisory process are presented to the customer taking into consideration his knowledge level and interests. The result of an advisory process is a set of identified products which fit to the needs and wishes of the customer. In order to guarantee the confidence in the result, a set of explanations for the identified solution is given.

Mass customization of services in the *public sector* is majorly triggered by the increasing structural complexity of society. On the one hand laws and regulations become more complex in order to consider different social interest groups on the other hand customers (i.e. citizens, entrepreneurs, etc.) are forced to deal with a much larger and complex set of services. While in the private sector tools for open innovation are increasingly provided for customers, in the sector of public administration first steps are set toward this direction. (Lenk et al., 2001) state that a great potential exist to improve the interface between administrative agencies and its customers[1]. Todays

[1]Throughout the paper we use the term customer to de-

Felfernig A., Russ C. and Wundara M. (2004).
TOOLKITS SUPPORTING OPEN INNOVATION IN E-GOVERNMENT.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 296-302
DOI: 10.5220/0002648402960302

e-government Web pages provide useful functionalities starting with simple presentations of organizational units and ending with different facets of online transactions (e.g. renewal of a driving license or a passport). Best practice examples provide life event oriented views on the given services (Kavadias and Tambouris, 2003). These approaches are useful to increase the usability of large e-government portals, however, none of those environments provides services for effectively and interactively guiding and assisting customers to find the services they require. An entrepreneur searching for financial support for research projects is forced to search through a large number of different documents describing the purposes of different research programs. Another entrepreneur trying to identify the best suited place of business in a certain region is in the exactly same situation. In such situations virtual advisors can help customers by providing personalized and explanatory dialogs with corresponding advisory results (e.g. well-suited places of business for an entrepreneur including an explanation why the recommended place was chosen). This advisory support is provided via *virtual advisory channels* (see Figure 1). A customer using this channel contacts a virtual advisor to retrieve services useful in his current situation (life event), i.e. the customer initiates a dialog with the advisor system.

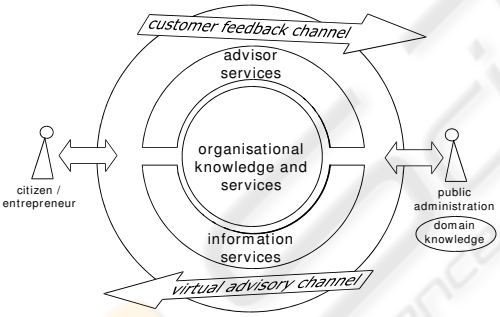

Figure 1: Open innovation in e-Government

Based on a set of stored user interactions stemming from the dialog with the virtual advisor, the domain expert (public administration in Figure 1) is now enabled to identify *weaknesses in the actual advisory process* (e.g. which customer requirements significantly correlate with the cancellation of the advisory process?) and to identify *customer requirements not considered up to now* (e.g. are there financial support programs frequently demanded by customers but not provided in the actual portfolio?). An interpretation of advisory sessions supports the domain expert when developing and maintaining advisor applications (this support is provided via *customer feedback channel* - see Figure 1). Feedback cycles are the basic idea behind the concept of *open innovation*, i.e. on the one hand to provide advisor functionality for customers to alleviate the access to a complex product or service, on the other to learn from customer preferences in order to improve, change and extend the product/service palette and the corresponding advisory processes. Such automated feedback cycles are extremely useful since *politicians* and *other decision makers* in public administration can detect and react much faster to open requirements existing in their community, furthermore flexible improvements of existing advisory processes are enabled.

The remainder of this paper is organized as follows. In Section 2 we sketch the different components of an open innovation environment in the context of public administration. Section 3 contains a description of the used technologies, finally in Section 4 we discuss related work. Section 5 concludes the paper.

2 SYSTEM ARCHITECTURE

Basically, an open innovation environment consists of the components sketched in Figure 2 - the role of these components and their interaction is discussed in the following subsections.

Knowledge Acquisition. In order to be up-to-date, the advisor knowledge base (product model, product data, user profiles, interaction data) is continuously adapted to the new situation (e.g. when new laws or other regulations are introduced). In this context a central modeling environment (Knowledge Acquisition component in Figure 2) is needed supporting the effective maintenance of the advisor knowledge base. When new laws are introduced the corresponding advisor system should already be available to the customer, i.e. the development- and maintenance time for the advisor system is strictly limited.

As constraint representation languages of advisor systems are hardly understandable for domain experts and even for programmers themselves, development environments for those systems must provide intuitive modeling concepts allowing the domain expert to effectively develop and maintain advisor applications. In Section 3.1 the design of advisor knowledge bases using the Unified Modeling Language and its integrated constraint language OCL (Object constraint language) (Warmer and Kleppe, 1999; Rumbaugh et al., 1998) is presented. UML is well known and frequently applied in industrial software development processes - the approach is intuitive and alleviates the integration of knowledge-based advisor

note citizens, entrepreneurs, lawyers etc. having to interact with public agencies.

techniques into industrial software development processes.

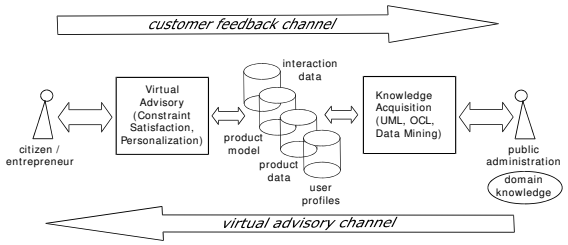

Figure 2: System architecture

Advisor knowledge bases consist of the following parts:

- Product model: this model represents the basic structure of the product including additional constraints (business rules) imposing restrictions on the usage of different (sub-)products. An example for such a product model is shown in the UML (Rumbaugh et al., 1998) model of Figure 3, additional constraints (business rules) are shown in Figure 4 and Figure 5.
- Product data: the product model contains a set of references to basic information sources which are presented to the customer as part of the result of the advisory process. If the result of a financial support advisor is a certain program (e.g. the IST program in Figure 3), additional information concerning this program is stored in the underlying product catalog (e.g. the attribute *advisor* contains a reference to a specialized IST program advisor).
- User profiles: each registered user is described by a set of dimensions describing his/her interests and his/her knowledge level concerning different service areas. This profile is used in order to personalize the advisory processes and the user interface.
- Interaction data: user interactions from advisory sessions are stored here for the purpose of analysis. Following the idea of open innovation domain experts are enabled to analyze already given customer interaction sequences. In order to support such feedback cycles the knowledge acquisition component includes *data mining* functionalities supporting the derivation of association rules and the application of different sorts of clustering mechanisms.

Virtual Advisory. Virtual advisors support user-centered dialogs by imposing questions and providing explanations - the result of an advisory session is a set of proposed solutions with corresponding explanations why this solution fits to the needs of the customer. Within the context of virtual advisor applications customers have different knowledge levels and interests concerning a set of provided services. For example, customer A is very well informed about the current situation w.r.t. to financial support of research projects, while customer B knows nothing about this domain and must get more detailed information about the available programs. Consequently the whole dialog with the customer must be organized in a personalized fashion considering different interest- and knowledge levels of customers w.r.t. the provided service palette. Depending on the customers answers and interaction behavior an intelligent user profile can be developed. Which questions, hints and answers are presented to the user, strongly depends on the information stored in this profile.

3 CONCEPTS OF VIRTUAL ADVISORS

3.1 Knowledge Representation

Our definition of an advisory task is based on the definition of a configuration task (Felfernig et al., 2000).

Definition: Advisory task. An *advisory task* is defined as a tuple consisting of the advisor knowledge base ADV_{KB} and a given set of customer requirements ADV_{REQ}, i.e. <ADV_{KB}, ADV_{REQ}>. ADV_{KB} consists of a description of product properties - concrete products (entries from a product catalog) are included in ADV_{KB} as well. Additionally, a set of *constraints* defines restrictions on allowed combinations of instantiations of product properties of ADV_{KB}. The second part of an advisory task is a given set of *customer requirements* ADV_{REQ} which represent a set of instantiations of given customer properties in the product model. □

Definition: Advisory solution. The goal of the advising process is to identify a solution ADV_{SOL} which fulfills the imposed customer requirements ADV_{REQ} where all the constraints of ADV_{KB} are fulfilled, i.e. $\text{ADV}_{REQ} \cup \text{ADV}_{KB}$ is consistent. □

Service descriptions of the advisor knowledge base are represented in UML (Rumbaugh et al., 1998) class diagrams (a simple example for such a service description is shown in Figure 3 which will be used as working example throughout the following sections). The basic service structure is modeled using classes (e.g. the class *FinancialSupport*), associations (e.g. the association between the class *Customer* and the

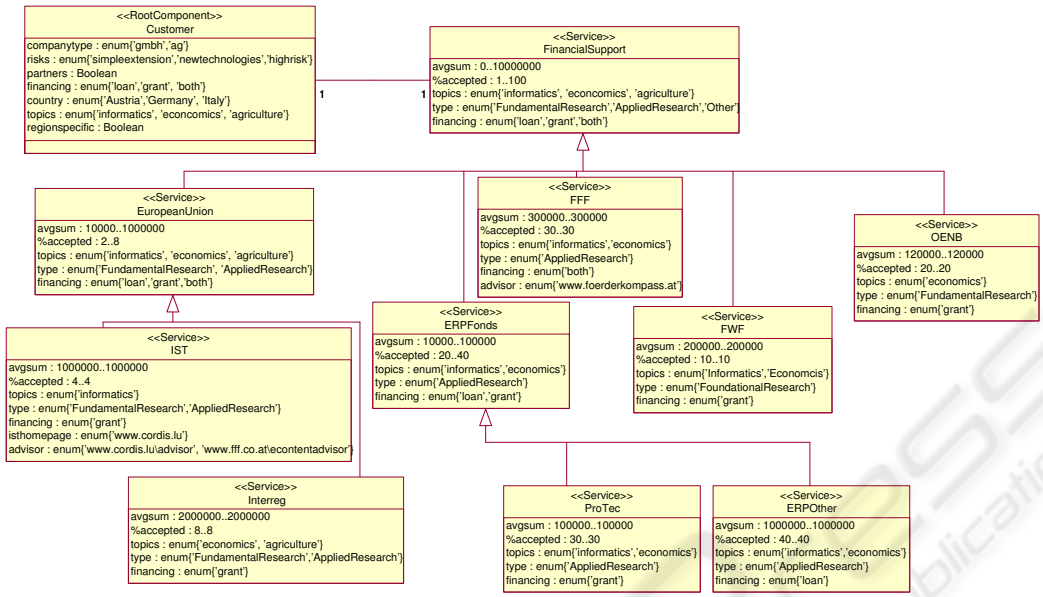

Figure 3: Service structure

class *FinancialSupport* which represents the relationship between a set of imposed customer requirements an the corresponding advisory solution ADV$_{SOL}$), and generalization hierarchies (e.g. *FFF* represents a special kind of *FinancialSupport*).

Constraints defining restrictions on allowed combinations of services are represented using the Object Constraint Language (OCL) (Warmer and Kleppe, 1999) which is an integrative part of UML. Figure 4 shows two example constraints imposed on the service structure shown in Figure 3. The first constraint denotes the fact that financial support for customers from foreign countries is restricted to cooperations within the framework of European Union (financial support of service type *EuropeanUnion*) projects. The second constraint denotes the fact that simple extensions to software components are not supported by the current types of financial support programs[2].

Basically, there are three different classes of constraints which are used very often when building knowledge bases for virtual advisors.

- Requirement constraints: in some cases, the existence of a specific customer requirement (e.g. country <> 'Austria') requires the existence of a specific service or a specific parametrization of a service in the advisory solution (e.g. *FinancialSupport.oclIsTypeOf*(*EuropeanUnion*)).
- Compatibility constraints: certain combinations of customer requirements (e.g. no project partners are available and the customer stems from a foreign country) can not occur in the same advisory session - either these requirements are directly incompatible or do not allow the identification of a solution for the given advisory task.
- Resource constraints: parts of an advisory task can be interpreted as a resource balancing task, where some classes in the knowledge base produce some resources and other classes act as consumers of resources. E.g. when providing a more complex financial support advisor, different potential partners of a project consume resources in the form calculated costs - these costs must no exceed the overall limit imposed by a given financial support program.

These three types of constraints represent frequently used model restrictions in advisor knowledge bases. For such types of constraints we provide a set of graphical interfaces (constraint schemes) in order to alleviate the constraint implementation for the developer of the advisor application. An example for a requirement constraint represented on the graphical level is shown in Figure 5. Note that constraints are bound to a corresponding explanation - e.g. if the customer has specified that (s)he is interested in local funding opportunities and has an enterprise with more than 500 employees, a corresponding hint could

[2]For reasons of readability the examples are kept simple and do not regard the whole complexity of financial support programs.

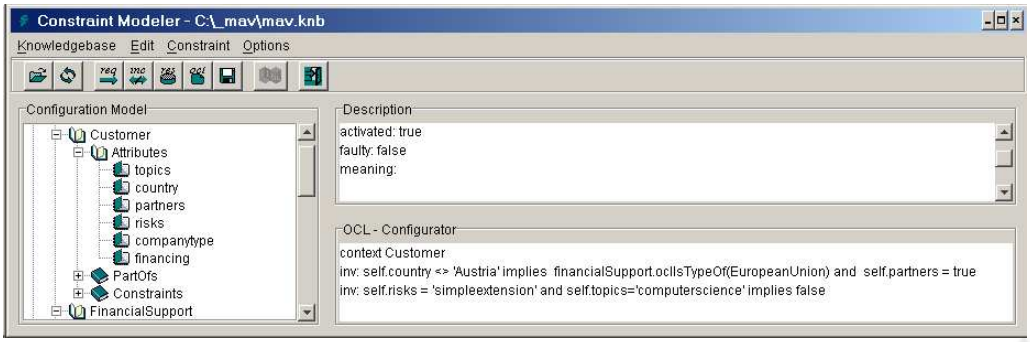

Figure 4: Example constraints

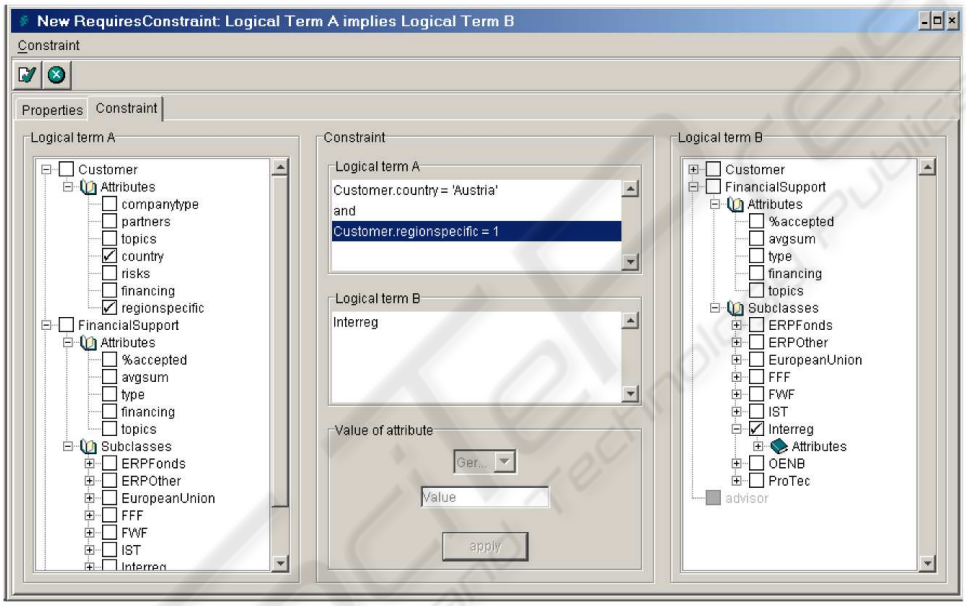

Figure 5: Graphical representation of requirement constraints

be presented indicating that no local funding is available for companies with more than 500 employees.

3.2 Personalization

As already mentioned, it strongly depends on the information stored in the user profile which questions, hints and answers are presented to the customer during an advisory session. Personalization concepts play here an important role since one cannot assume that the customer is well acquainted with the provided services. On the basis of given interaction traces a customer can be described using a set of abstract dimensions related to the provided products and services (R. Schaefer, 2000).

Lets have a short look at the financial planning domain. Investment products can be categorized w.r.t. to the dimensions *profit*, *availability*, and *risk*. Depending on the given customer requirements we can determine a certain distribution describing the interest of the customer w.r.t. the given dimensions. If the customer articulates that no financial reserves are available in the current situation, the dimension risk will be of more importance, i.e. products with no financial risks will be primarily presented to the customer. A distribution of (*profit*=30, *availability*=60, *risk*=10) describes a customer with nearly no readiness to accept risks, with high interest in availability of the investment and also an increased interest in profitableness of the investment. The calculation of such distributions is based on scoring mechanisms, i.e. depending on customer answers a certain amount

of points is added to each dimension [3].

Regarding our financial support scenario we introduce the dimensions *support* (amount of financial support which can be assumed if the project proposal will be accepted) and *acceptance* (the probability of the acceptance of the project proposal). For a concrete customer let us assume the following distribution {support: 30, acceptance: 60}, i.e. the customer is much more interested in getting the project proposal accepted than in getting a high amount of financial support. Furthermore, let us assume that the virtual advisor has determined four solutions $\{S_1, S_2, S_3, S_4\}$ for a given set of customer requirements (representing a request). For this set of solutions and the given distribution {support: 30, acceptance: 60}, a personalized utility $g(x)$ can be calculated for each solution alternative using the formula $g(x) = \sum_{i=1}^{n} e_i(x)s_i(x)$.

For each solution it is determined to what extend this alternative is relevant for the different dimensions. The utility $g(x)$ is defined by $s_i(x)$ (relevance of the alternative for the dimension i) and by $e_i(x)$ which represents the customer interest in the dimension i. The value $g(x)$ for S_1, i.e. $g(S_1)$ = 30 * $s_{support}(S_1)$ + 60 * $s_{acceptance}(S_1)$. This formula has to be applied for each of the given solution alternatives - the result of the calculation is a customer-specific ranking of the alternatives, e.g. S_1, S_3, S_2, S_4. This ranking of advisory results is one possible application of the presented personalization concepts. The same concepts are used within the scope of the advisory process in order to determine the utility of different alternatives for questions, hints or explanations. Furthermore, these concepts can be used in order to determine default settings for parameters which otherwise would have to be explicitly provided by the customer. This is helpful when a customer does not know very much about the service domain or a parameter is of no interest for him.

3.3 Data Mining

In the context of open innovation, data mining techniques (Berry and Linoff, 2000) can be applied in order to determine patterns from previous interaction sequences in the form of e.g. association rules or clusters. The result of such an analysis can be used for two different purposes. On the one hand the initial version of an advisor application is not the last resort, in many cases insufficient explanations and wrong results create an increasing frustration on the part of the customer. In this context data mining techniques can be applied for detecting areas in the advisor knowledge base which are responsible for the bad system behavior. On the other hand customer requirements can be analyzed with the goal to detect so called hidden requirements, i.e. requirements often articulated by customers but not supported by the given service palette. A simple example for such a result could be that computer science projects proposed by institutions from foreign countries[4] having no domestic partners are not supported by domestic financial support programs, i.e. *topics='informatics' and country <> 'Austria' and partners = false ==> 'no solution'*. In the following the question has to be answered whether this is intended or whether there exist opportunities to introduce such a financial support program (maybe on an intra-regional level) in the future.

4 RELATED WORK

The *Webocrat* portal system proposed by (Paralic and Sabol, 2001) supports customer integration by providing knowledge management (discussion forum), content management (Web content management), and polling functionalities (opinion polling room). The citizen information helpdesk is a layer integrating these functionalities by providing a text-based interface supporting the search for support material. The system supports personalization functionality by allowing the user to declare his/her topics of interest. Systems like Webocrat provide a first view on the design of future e-government systems.

(Vamos and Soos, 2003) discuss AI approaches in the area of natural language understanding and case-based reasoning bringing public administration closer to its customers. This work can be seen as complementary to the concepts presented in this paper and will be taken into consideration for future inclusion into the presented toolsuite.

(Kavadias and Tambouris, 2003) present GovML, a markup language to define structures for governmental data and metadata. Compared to our work GovML provides a domain-specific vocabulary for the representation of public services and life events, whereas our approach uses the concepts provided by a domain-independent modeling language. The interesting point here is how to integrate the concepts presented in GovML into a standard UML profile for the area of public administration.

(Lenk et al., 2001) point out the great potential which exists to improve the interface between administrative agencies and the citizens. This interface ranges from a mere human mediated form of interaction to intelligent software agents. These agents immediately provide information to users on where to get online help and active support in specific cases.

[3] A more detailed discussion on how to determine such a distribution can be found e.g. in (R. Schaefer, 2000).

[4] In this example countries <> 'Austria' are interpreted as foreign countries.

In this context our work is positioned on the level of intelligent software agents providing intelligent information and advisory services to customers.

In (Baar et al., 2000) experiences from the application of OCL in industrial software development processes are discussed. In principle, OCL seems to be quite useful and software engineers and even domain experts with a technical background are able to apply OCL for stating formal constraints on given object models. Especially software engineers accepted OCL because of the similarities of its syntax to object-oriented programming languages. However, (Baar et al., 2000) point out that more intuitive concepts are needed in order to support effective modeling of OCL constraints. In order to tackle this challenge (Baar et al., 2000) introduce the notion of constraint schemes which are parameterizable constraints which can be differently instantiated depending on the actual situation. For example, a constraint schema could restrict the number of objects of a class to an upper bound - this upper bound is represented by a variable which must be instantiated in order to instantiate the corresponding OCL constraint. The graphical input masks presented in this paper are very similar to the constraint schemes presented by (Baar et al., 2000).

5 CONCLUSIONS

In this paper we have presented a toolsuite supporting the concept of open innovation in e-government. Such a toolsuite allows customers (citizens, entrepreneurs, lawyers, etc.) to articulate their requirements, points out and explains inconsistencies in given customer requirements and provides needed service packages as outcome of the advisory process. Using tracking data from advisory processes, public agencies are enabled to detect problem areas in advisor knowledge bases and to generate innovations that meet the needs of their customers. The construction of Virtual Advisors requires a corresponding knowledge acquisition workbench which allows the effective construction of advisor knowledge bases. We have presented such a workbench which is based on the Unified Modeling Language (UML) and the integrated Object Constraint Language (OCL).

REFERENCES

Anderson, D. (1997). *Agile Product Development for Mass Customization*. McGraw-Hill.

Baar, T., Haehnle, R., Sattler, T., and Schmitt, T. (2000). Entwurfsmustergesteuerte Erzeugung von OCL Constraints. In Mehrhorn, K. and Snelting, G., editors, *Informatik 2000, 30. Jahrestagung der Gesellschaft fuer Informatik*, pages 389–404. Springer-Verlag.

Berry, M. and Linoff, G. (2000). *Mastering Data Mining Techniques*. John Wiley & Sons.

Felfernig, A., Friedrich, G., Jannach, D., and Stumptner, M. (2000). Consistency-Based Diagnosis of Configuration Knowledge Bases. In *Proceedings of the* 14^{th} *European Conference on Artificial Intelligence (ECAI 2000)*, pages 146–150, Berlin, Germany.

Kavadias, G. and Tambouris, E. (2003). GovML: A Markup Language for Describing Public Services and Life Events. In Wimmer, M., editor, *Knowledge Management in Electronic Government: 4th IFIP International Working Conference, KMGov 2003, Rhodes,Greece, May 26-28, Lecture Notes in Computer Science*, volume 2645, pages 106 – 115. Springer-Verlag.

Lenk, K., Traunmueller, R., and Wimmer, M. (2001). Electronic Business Invading the Public Sector: Considerations on Change and Design. In *34th Hawaii International Conference on System Sciences (HICSS-34)*, volume 5. IEEE.

Paralic, J. and Sabol, T. (2001). Implementation of e-government using knowledge-based system. In *DEXA Workshop*, pages 364–368.

Piller, F. and Stotko, C. (2003). *Mass Customization und Kundenintegration*. symposion, Duesseldorf.

PineII, B., Victor, B., and Boynton, A. (1993). Making Mass Customization Work. *Harvard Business Review*, Sep./Oct. 1993:109–119.

R. Schaefer, M. B. (2000). Ein allgemeiner Ansatz zur Personalisierung von Webseiten mit Informationsdiensten. In *Adaptivitaet und Benutzermodellierung in interaktiven Softwaresystemen*.

Rumbaugh, J., Jacobson, I., and Booch, G. (1998). *The Unified Modeling Language Reference Manual*. Addison-Wesley.

Tseng, M. and Piller, F. (2003). *The Customer Centric Enterprise*. Springer, Duesseldorf.

Vamos, T. and Soos, I. (2003). Data Management and AI in E-Government. In Wimmer, M., editor, *Knowledge Management in Electronic Government: 4th IFIP International Working Conference, KMGov 2003, Rhodes,Greece, May 26-28, Lecture Notes in Computer Science*, volume 2645, pages 230 – 238. Springer-Verlag.

VonHippel, E. (2001). Perspective: User Toolkits for Innovation. *Journal of Product Innovation Management*, 18:247–257.

Warmer, J. and Kleppe, A. (1999). *The Object Constraint Language - Precise Modeling with UML*. Addison Wesley Object Technology Series.