How Knowledge Management Can Support the IT Security of eGovernment Services

Markus Nick, Stephan Groß, and Björn Snoek

Fraunhofer Institut Experimentelles Software Engineering (IESE)
Sauerwiesen 6, 67661 Kaiserslautern, Germany
{nick, gross, snoek}@iese.fhg.de

Abstract. Safeguarding security for eGovernment services is an essential ingredient for the success of such services. For this purpose, isolated security efforts are not sufficient. Integrated concepts are required. In the publicly funded project SKe, we are developing such an integrated approach. One component of this integrated approach is a knowledge management-based solution to support the dynamic aspects of IT security. The component - an intelligent IT security console - supports the daily work of the IT security personnel and supports them in systematically recording and using experiences in their work process. The component is being developed in cooperation with an application partner and is also used in projects with industrial partners.

1 Introduction

eGovernment is becoming a more and more important issue in a number of countries. For example, in Germany there is the initiative BundOnline 2005 [9] that aims at making all services of the federal government (German: "Bund") online until 2005. In the USA, eGovernment is regarded as the next American Revolution that is expected by the public [24]. All these initiatives have in common that "safeguarding security and privacy is the top priority of the public for eGovernment services".

In the following, we refer with the term *eService* to such eGovernment services. An eService is the electronic counterpart of an interactive procedure between governmental organizations and their customers.

To ensure the security of eServices, integrated security concepts are needed that do not only address the technical issues of IT security but also relevant organizational, cultural, and social aspects as well as legal peculiarities of the implemented administrative procedure [25]. Furthermore, the security level has to be reasonable, i.e., a good solution has to be found that satisfies the often-conflicting goals of security and usability. Therefore, the success of modern eGovernment services dramatically depends on trustworthiness. So, reasonable security has to be ensured, which is one of the bases for user acceptance [18].

All this is in line with recent developments in the field of IT security, where a major change in the way of handling IT security happens. The IT security expert Bruce Schneier phrases this as "IT security is a process and not a product," which means that it is not sufficient to install a number of products to secure the IT of an organisation [21]. Case-based reasoning is a principle and technology that has the potential to enrich such processes [16].

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-540-44836-5_33
M.A. Wimmer (Ed.): KMGov 2003, LNAI 2645, pp. 151–162, 2003.

Unfortunately, today's IT security concepts are not yet integrated as required for eServices. They cover only individual aspects and, furthermore, their relationships are not clarified. There is also a lack of (a) clarity about formal conclusiveness of a security concept, (b) the correctness of implemented security measures, (c) continuous monitoring of the measures, and (d) systematic recording of security incidents and experiences.

The presented work is part of a comprehensive approach being developed in the project SKe. The topic of SKe is to develop integrated security concepts and mechanisms for continuously ensuring and improving the required security levels during operation [20, 1]. In SKe, formal modelling of the eService process and its security aspects is used for identifying and verifying the required security properties of an eService [22]. However, to ensure the security requirements, the models require explicit preconditions and assumptions to be fulfilled. To ensure these preconditions and assumptions, a set of technical and organisational measures is derived. Another component of SKe, the electronic security inspector (eSI) supports the continuous monitoring of security measures that can be checked software-technically. By also collecting the experiences on the reaction on security incidents (events, breaches, threats, etc.) in an experience-based security database (eSDB or "experience base") and making them available through an intelligent IT security console, we provide fast and sophisticated support for the daily security-related work. Though the results of SKe can be applied to non-governmental eServices as well, the project is especially adapted for eGovernment services. For example, we cooperate with a governmental unit responsible for a financial eService of a major German city as application partner. The partner is involved in the development of the IT security console and will use the system in the context of a case study for the project.

In this paper, we focus on the intelligent IT security console and the experience-based security database (eSDB). Besides the development of the experience base, an overall process for the required security and knowledge-based activities has to be defined [4]. An integrated knowledge model ties together the formal security model with the experience-based security database.

The solutions for IT security console and eSDB have to be flexible and scalable with respect to different eServices and, therefore, different organisational infrastructures and different experience structures. This leads to different needs regarding "intelligent" support from the knowledge management (KM) system. Furthermore, a tight integration into the work process is required to make KM successful, i.e., a proactive, context-sensitive delivery of knowledge within the work process [3]. We use case-based reasoning (CBR) [2] as a principle and technology for knowledge management. Regarding CBR, the methods and technologies for IT security console and eSDB are also related to CBR-based diagnosis [12, 14], Textual CBR [13], and CBR maintenance [11, 19]. Furthermore, the work is based on our own methods and tools for experience base and CBR maintenance and evaluation [16, 17, 5]. The experience factory concept serves as organisational principle [6]. By establishing a feedback cycle and supporting this feedback cycle with the IT security console, we integrate the recording and usage of experience into the daily work process of the IT security personnel. With the comprehensive view on environment, knowledge model, and processes, intelligent IT secu-

rity console and eSDB become typical applications of the rather new field of *experience management (EM)* [7, 23, 5].

The expected benefits of KM for maintaining the IT security of eServices are manifold: We expect to establish a feedback cycle for continuous learning and improvement of the IT security of eServices based on experiences. An experience-based security database (eSDB) is expected to speed up and improve the reaction to security threats and incidents. Furthermore, an eSDB is expected to improve the traceability regarding standard cases and non-standard cases. In addition, the systematic recording of incidents in the eSDB provides a good basis for preparing security audits. Last but not least, the systematic recording of experience allows maintaining a certain minimal acceptable level of security even when IT security personnel are not available (e.g., on holiday, illness, fluctuation).

The remainder of the paper is structured as follows: Section 2 describes the application of the KM method DISER [23, 5] for identifying the scenarios that are relevant for managing IT security experience. This resulted in the focus on experience on the reaction of security incidents and -as the core scenario- an experience feedback loop, which integrates the recording and usage of experience into the daily work of IT security personnel (Section 3). To translate this feedback loop into action, we are developing an intelligent IT security console in close cooperation with our application partner (Section 4). The plans for the evaluation of the IT security console are summarized in Section 5. The paper closes with a summary and conclusion (Section 6).

2 KM for IT Security in eGovernment Solutions

We applied the KM method DISER [23, 5] to develop a vision for KM support and CBR support for IT security personnel for eServices. Detailed results are documented in one of the SKe project deliverables [4].

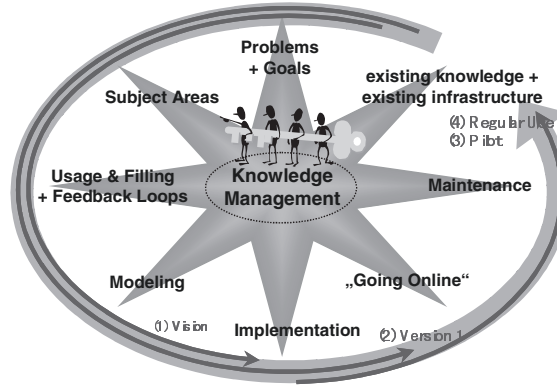

Fig. 1. An overview on DISER (=Design and Implementation of Software Engineering Repositories)

In Phase 1 of DISER, starting with the major goal (as stated above) and existing knowledge such as the *baseline protection manual* as a kind of German standard from the BSI [8], we identified four subject areas. For these, an overall process for the required security and knowledge-based activities was developed [4], which consists of 14 scenarios and an integrated knowledge model that ties together the formal security model with the eSDB. The scenarios were mainly derived from or based on the IT security process from the baseline protection manual [8], and scenarios elicited in workshops with our project partners [4].

For the eSDB, this resulted in a so-called *experience feedback loop* for the systematic recording and application of IT security experience about the reaction on security incidents and 2 scenarios refining the feedback loop. For the 2nd Phase of DISER, which has the objective of systematically developing the design of the envisioned KM system and building a first version, this feedback loop was chosen as the core scenario. In the following, we focus on the experience feedback loop and its refining scenarios as basis for the IT security console.

3 Feedback Loop for Dynamic Controlling of IT Security

The feedback loop from the viewpoint of the IT security personnel is depicted in Fig. 2. We distinguish four phases in the loop: security status monitoring, diagnosis & decision support, reaction, and feedback. While the monitoring of the security status is a quasi-continuous task in order to keep the security measures effective, the other three phases are triggered by the recognition of potential security incidents and run sequentially for each of these incidents.

In the *security status-monitoring* phase, the electronic security inspector (eSI) monitors software-technically checkable objects using so-called sensors. The organisational security inspector (oSI) collects respective data on organisational measures or on measures that cannot be monitored by the eSI for other reasons. All states that cannot be classified as "OK" by eSI or oSI (i.e., people) are compiled in a list of potential security incidents. In the *diagnosis and decision support* phase, the status is determined for "unclarified" cases. Then respective reactions are proposed based on the experiences and selected by the person(s) responsible for IT security. Depending on severity and potential damage, a priority is assigned to each incident. The incidents to examine are put on a to-do list. In the *reaction* phase, the items on the to-do lists are handled by the respon-

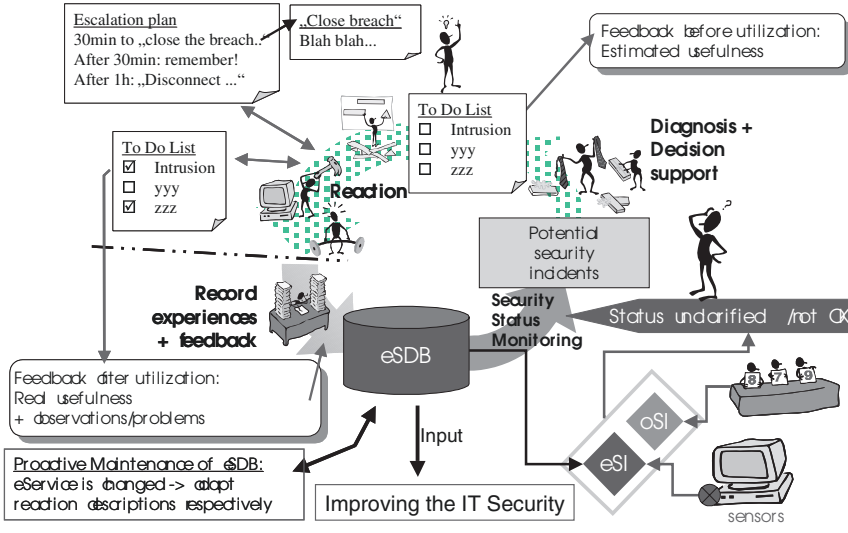

Fig. 2. A feedback loop for IT security experience, which is integrated into the work process.

sible IT security persons according to the priorities. Furthermore, regarding the escalation hierarchy for highly critical situations, automatic reactions are considered if the responsible persons are not available and do not or cannot react quickly enough (e.g., switching off the internet connection when eSI reports a breach and no-one is available because of a public holiday). In the *feedback* phase, experience on new or different reactions is recorded and feedback is given, e.g., if there was really a security incident, effect and success of the reaction, actual damage, and prevented damage. This feedback and new experience closes the loop by improving the diagnosis and decision support capabilities.

When an eService is changed, the proactive maintenance supports the identification of the relevant cases that have to be updated.

The systematic recordings on the security incidents can be used for identifying the need for improvements of the IT security and, as a consequence, introduce new measures or upgrade existing measures.

The feedback loop is an instantiation of the standard case-based reasoning cycle [2] at the organisational level [23] and shows a tight integration into the work process, which is an enabler for successful knowledge management [3].

4 An Intelligent, Experience-Based IT Security Console for eServices

The IT security console implements the core component of the SKe process, i.e., the feedback loop for IT security experience and its related scenarios. Its core task is the support of the IT security personnel in their daily work.

To ensure intelligent support within the daily work context of the user, we developed a process model and a graphical user interface that completely integrates the intelligent support (*iSupport*) into the work process (Section 4.1). This process model is based on the feedback loop from the previous section. To store experience in a standardized way, a representation schema is necessary to describe how the knowledge is recorded (Section 4.2). The schema allows distinguishing between standard and non-standard reactions on incidents. The maintenance process supports the merging of standard cases and related non-standard cases. The strategy for the maintenance process is adaptable to the needs of the environment (Section 4.3). Finally, the actual implementation is based on a product line architecture for experience management systems (Section 4.4).

4.1 How the IT Security Console Supports IT Security Personnel

We use a process model to describe how the IT security console supports IT security personnel. The process model is depicted in Fig. 3 using states (boxes) and transitions (arrows). Based on the process model, we developed a graphical user interface (GUI) for the IT security console. In the following, we describe the process and the opportunities for intelligent support in the different steps of the process and illustrate this with examples from the GUI.

After the login, the system presents the user his personal to-do list and the list of new, open incidents. An IT security manager can additionally view a list of all incidents in the to-do lists of the IT security specialists. New incidents can be reported by eSI (e.g., a port scan) or oSI or entered manually by the IT security personnel.

When a new incident in the list of potential incidents is assigned to a person by himself or by the IT security manager, further situation characteristics are added, which can not be determined automatically, e.g., cause. An iSupport proposes a set of categories. Another iSupport proposes the priority level based on similar existing cases. A third iSupport proposes reactions on the incident based on same or similar cases in the past. The user selects the reactions he regards as most useful for dealing with the new incident.

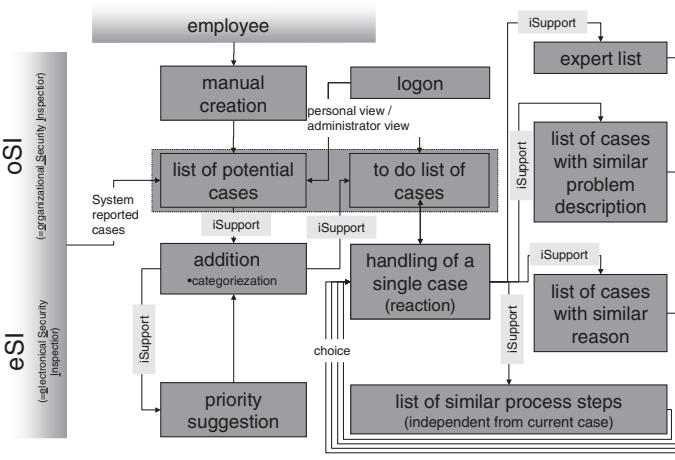

Fig. 3. The process underlying the IT security console

Prioritisation is relevant when there are more than 5-10 tasks in the to-do lists in the typical case for a certain environment. According to our experience from industrial projects, priority decisions are subjective and cannot be made consistently over a longer period of time neither by a single person nor by several persons. Therefore, iSupport standardizes the priority decisions.

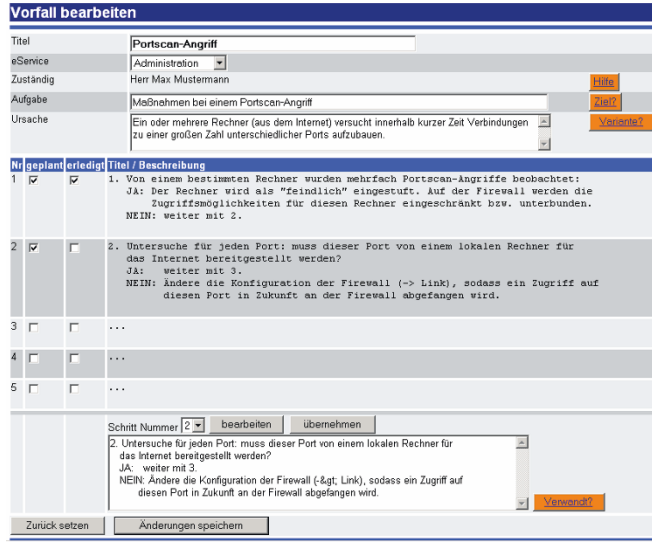

Fig. 4. Part of the GUI with the steps of the reaction. The orange buttons are for the iSupports "Help", "Goal", "Variant", and "Relatives".

To address the requirements of different environments (different eServices, organisational structure, and experience structure), we developed different variants for the presentation and model-

ling of the description of the reaction, which differ in their functionality and degree of formality. Based on example cases from and in discussions with our application partner, we identified structured text as the most adequate basis for representing the reaction. Variants of structured text for the reaction range from a simple single text field over text with a structure for steps to a decision tree-like structure. The representation of steps is necessary to provide support for planning and executing a reaction that takes longer to perform. The variants also differ with respect to their support for parallel execution. A further formalization is required for steps that can be performed automatically. Fig. 4 gives an impression of the resulting variant selected by our application partner, which supports parallel processing of single steps by the IT security personnel.

In the reaction phase, the system provides several iSupports, which are launched by simply pressing a button (see Fig. 4):

- *"Help" (Gr. "Hilfe")* supports persons who have further questions or who do not completely trust the answers of an IT system. As support, a number of experts for the current incident and reaction are identified. This is done by retrieving a list of similar cases from the experience base and presenting an overview of these cases together with the contact information of the person who handled the respective case in the past.
- *"Goal" (Gr. "Ziel")* aims at finding alternative reactions for solving the problem. For this purpose, a list of cases with same or similar cause and task is retrieved.
- *"Variant" (Gr. "Variante")* aims at identifying alternative causes for the problem. For this purpose, a list of cases with same or similar tasks is retrieved. When an alternative cause is identified, its reaction is also proposed as solution for the current case.
- *"Related" (Gr. "Verwandt")* aims at finding related information for a single step. For this purpose, the systems searches for cases and steps that are similar to the current step (e.g., in the port scan case for the step where the necessity for "open" ports at the firewall is checked, firewall configuration experience would we found). This iSupport requires a representation that supports steps.

The described process refines the feedback loop with respect to the user interaction of the IT security console and shows the integration into the work process as outlined by the feedback loop.

4.2 Modelling the Experience

To support the process model described in the previous section, the cases are structured according to the schema as depicted in Fig. 5. For the design of the schema, we had to consider the iSupports, feedback, and maintenance of standard cases (as a dynamic handbook for reactions on security incidents). Furthermore, the structure has to be open for further functionality such as version management.

The schema distinguishes between *standard cases* ("case", "step-position", and "step") and *concrete cases* ("case occurrence" and "step occurrence"). While standard cases provide a mature description for the reaction in a certain situation, concrete cases describe the application of standard cases for a concrete incident as well as non-standard cases. The application of a standard case can differ regarding the order of the execution

of the steps, etc. Such concrete cases are accumulated over time for a standard case [15]. In a maintenance cycle, these concrete cases are used for improving the standard cases. This implements an experience-based improvement cycle for standardized reactions.

The core of a concrete case is the "case occurrence" that includes the attributes of the actual reaction in the concrete case. Besides the "editor" (the responsible IT security person), also "controller" and date & time are stored. The controller is usually the responsible IT security manager. A yellow note allows the editor to store some notes during the processing of the case (e.g., to whom a subtask was assigned). For our application partner, a yellow note is deleted after closing a case. Besides the general situation attributes (trigger, task, title), the steps of the reaction are modelled. The steps themselves are described textually. For the concrete case, the planning and execution is supported at the step level by providing a respective status for each step.

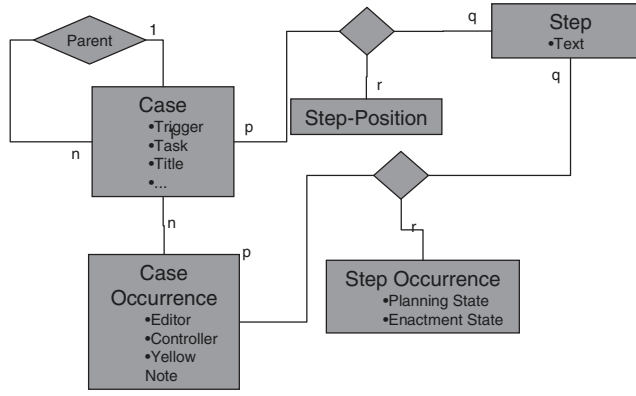

Fig. 5. Schema for experience on IT security incidents and reaction

4.3 Maintenance Processes and Strategies

The maintenance processes deal with the question of when to update standard cases. For this purpose, there are two classes of concrete cases to be considered: (1) concrete cases that describe the unchanged application of the standard case, where the number of these cases indicates the validity of the standard case; (2) concrete cases that describe an application of a standard case with changes. In the following, we describe the options for the maintenance strategy and the meaning of "changed", i.e., when is a concrete case considered to have a "changed" reaction.

Basically, we distinguish between two extremes of maintenance strategies. The open and flexible nature of the described schema supports the range of maintenance strategies that is spanned by these two extremes:

1. Managed strategy: Changes to a proposed reaction are only recorded within the "concrete case". The standard case is not changed.
2. Unmanaged strategy: Each time when a proposed reaction is changed, a new standard case is recorded after finishing the handling of the incident. The new standard case is related to the old standard case via the parent relationship.

The managed strategy is the preferred strategy for handling cases where the standardization of the reaction is the major goal - for example, when the IT security personnel has a rather low level of expertise. In this case, a periodical or evaluation-triggered maintenance by the respective IT security expert is required [16]. This expert decides

which histories are integrated in updated standard cases or lead to new variants of standard cases.

The unmanaged strategy is preferred when the IT security personnel are mainly experts and have a high level of expertise. However, this leads to a high number of new standard cases when changes are made frequently. Furthermore, unchecked new standard cases can also lower the quality of the experience base when error-prone cases are added as standard cases.

In practice, an intermediate strategy is expected to be most useful. For this purpose, evaluation-triggered maintenance is used with a triggering by, e.g., the number of available concrete cases with changes in the reaction (compared to the reaction description of the respective standard case) or changes in the execution order of the steps of the reaction. For this purpose, only semantical changes count as changes; e.g., syntactical corrections of text do not count. The editorial work of the expert prevents a decrease in the quality of the experience base. For the revision, the expert can also analyse differences in the order of the execution of the different steps. The different options for the triggering of the maintenance allow a flexible response to the maintenance needs of the different environments.

4.4 Architecture

The system's architecture is an instantiation of IESE's experience base product line architecture INTERESTS[1]. The product line architecture uses a stable relational data base system as basis. On top of the database management system, we have application logic and user interface. Application logic and user interface are enhanced with intelligent components. Scalability addresses components and database schema. The schema has to support the scalability right from the start, e.g., when replacing a simple "intelligent" search with a more advanced solution. The scalability addresses features and also the price. For example, for the database system, the product line considers MS Access as inexpensive commercial-off-the-shelf tool, PostgreSQL as open source product, and Oracle as advanced solution; for advanced intelligent search, the commercial case-based reasoning tool orenge from empolis is used as an advanced component [10]. INTERESTS itself also delivers the specific glue for "gluing" together the components with program code as well as the respective knowledge on how to model and how to plug the components together.

For the application partner, an inexpensive solution is being developed. This solution is based on MS Access as database management system, J2EE-based technologies for the user interface (i.e., JavaServerPages and JavaBeans), and our in-house CBR solution for similarity-based retrieval on structured text. An excerpt of the GUI is shown in Fig. 4.

---

[1] INTERESTS = Intelligent Retrieval and Storage System.

5 Plans for Evaluation by Case Studies

For the evaluation, we distinguish three phases according to [17]. In the beginning of the usage of the system (i.e., Phase 1), we use our standard model for measuring indicators about the acceptance of the system [5, 16]. We combine measuring the usage of the system (i.e., number of queries per iSupport) and feedback on the utility of the retrieved experiences (i.e., the proposed reactions). Combining usage and utility allows obtaining a picture on the acceptance more quickly than just monitoring usage because -in the beginning- usage can also be high because the system is new and everybody plays with it. Furthermore, the utility feedback helps to obtain a better understanding of the users' real interests. Later, in Phase 2, application-specific issues can be added for a more detailed evaluation. Phase 3 focuses on the economic value of the system. For the IT security console, we are in Phase 1 and, therefore, will measure usage and analyse the utility feedback. These measurements will show if the feedback loop works.

6 Conclusion

We identified adequate IT security as an essential for eGovernment applications. In the SKe project, a comprehensive solution for ensuring the IT security of eGovernment applications is being iteratively developed and tested. Managing IT security knowledge is an integral part of the SKe approach. We are developing a so-called intelligent IT security console with special support for eGovernment web services as a means to use knowledge management for improving the IT security of eGovernment applications. This system supports the systematic collection and application of experiences on reactions to security incidents as well as the identification of standard cases and the proactive maintenance of the reaction in standard cases. Using case-based reasoning [2] as principle and technology, the system is able to provide intelligent support for several issues in the process of handling security incidents based on experiences in same or similar cases. For the system, we assured the integration of the intelligent support in the context of the work process [3] and developed a flexible schema and scalable architecture to allow an easy adaptation to the needs of different environments, i.e., eServices, organisational structure, etc.

The status of the work is as follows: Based on a design study using real-world cases from the application partner, the concept and user interface were reviewed by the application partner. Based on the design study, the most adequate variant for the representation of incidents and reactions as well as relevant opportunities for intelligent support were selected. A first version of the IT security console is being implemented and will be "fielded" in the first quarter of 2003. An evaluation program to demonstrate the usefulness of the solution will accompany the fielding. Furthermore, for the security department of a telecommunications company, we are coaching the development of a similar system from a simple case processing to a "real" KM solution.

Regarding the requirements and expected benefits, we draw the following conclusions: The role model supports the flexibility regarding the organizational structure. The IT security console can be adapted to different environments regarding the need for intelligent support (e.g., intelligent prioritisation is relevant for environments with a

high frequency of incidents). For the modelling of the reactions, we identified for the application partner a rather simple variant that allows the planning of single steps and storing the execution status of each single step. However, for industrial partners, we are developing a more complex modelling for the reaction, which contains a solution log ("story") with positive and negative experiences. The schema supports different strategies for the maintenance of the reactions on standard cases/incidents. These maintenance strategies range from unmanaged to managed style. This supports the expected benefits regarding the traceability of reactions on standard and non-standard cases. The scalable architecture allows an inexpensive start. The evaluation program for the first version will show if the IT security console is able to establish the experience feedback cycle in practice. So far, representatives of the intended users expect a more efficient and effective handling of problems with the financial eService and are looking forward to the first version.

The next steps are the finalization of the first version and a case study with the application partner to show that the IT security console provides the benefits in the practical application. For the second version, we will focus on the proactive maintenance of the reaction descriptions regarding changes to the eService. Encouraged by the good feedback from the application partner and other projects, we expect that knowledge management can provide a number of benefits for ensuring the IT security of eServices in eGovernment.

Acknowledgements. We would like to thank the German Ministry of Education and Research (BMBF) for funding the SKe project (contract 01AK900B). Furthermore, we would like to thank our colleagues at Fraunhofer IESE, the project members from Fraunhofer SIT, TU Darmstadt, and from the knowledge-based systems group at the University of Kaiserslautern for the fruitful discussions and their support.

References

[1] SKe - Durchgängige Sicherheitskonzeption mit dynamischen Kontrollmechanismen für eService Prozesse. http://www.ske-projekt.de/, 2001.

[2] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AICom - Artificial Intelligence Communications*, 7(1):39–59, Mar. 1994.

[3] A. Abecker and G. Mentzas. Active knowledge delivery in semi-structured administrative processes. In *Knowledge Management in Electronic Government (KMGov-2001)*, Siena, Italy, May 2001.

[4] K.-D. Althoff, S. Beddrich, S. Groß, A. Jedlitschka, H.-O. Klein, D. Möller, M. Nick, P. Ochsenschläger, M. M. Richter, J. Repp, R. Rieke, C. Rudolph, H. Sarbinowski, T. Shafi, M. Schumacher, and A. Stahl. Gesamtprozess IT-Sicherheit. Technical Report Projektbericht SKe - AP3, 2001.

[5] K.-D. Althoff and M. Nick. *How To Support Experience Management with Evaluation - Foundations, Evaluation Methods, and Examples for Case-Based Reasoning and Experience Factory*. Springer Verlag, 2003. (to appear).

[6] V. R. Basili, G. Caldiera, and H. D. Rombach. Experience Factory. In J. J. Marciniak, editor, *Encyclopedia of Software Engineering*, volume 1, pages 469–476. John Wiley & Sons, 1994.

[7] R. Bergmann. Experience management - foundations, development methodology, and internet-based applications. Postdoctoral thesis, Department of Computer Science, University of Kaiserslautern, 2001.
[8] Bundesamt für Sicherheit in der Informatikstechnik (BSI). *IT Baseline Protection Manual*. Oct. 2002. http://www.bsi.bund.de/.
[9] Bundesministerium des Inneren (BMI). Die eGovernment-Initiative BundOnline 2005. http://www.bundonline2005.de/, 2000.
[10] empolis GmbH. orenge (open retrieval engine) - empolis knowledge manager. http://www.empolis.comm/products/prod_ore.asp, 2000.
[11] D. B. Leake, B. Smyth, D. C. Wilson, and Q. Yang, editors. *Computational Intelligence - Special Issue on Maintaining CBR Systems*, 2001.
[12] M. Lenz, H.-D. Burkhard, P. Pirk, E. Auriol, and M. Manago. CBR for diagnosis and decision support. *AI Communications*, 9(3):138–146, 1996.
[13] M. Lenz, A. Hübner, and M. Kunze. Textual CBR. In M. Lenz, H.-D. Burkhard, B. Bartsch-Spörl, and S. Weß, editors, *Case-Based Reasoning Technology — From Foundations to Applications*, LNAI 1400, Berlin, 1998. Springer Verlag.
[14] L. Lewis and G. Dreo. Extending trouble ticket systems to fault diagnostics. *IEEE Network*, Nov. 1993.
[15] M. Nick, K.-D. Althoff, T. Avieny, and B. Decker. How experience management can benefit from relationships among different types of knowledge. In M. Minor and S. Staab, editors, *Proceedings of the German Workshop on Experience Management (GWEM2002)*, number P-10 in Lecture Notes in Informatics (LNI), Bonn, Germany, Mar. 2002. Gesellschaft für Informatik.
[16] M. Nick, K.-D. Althoff, and C. Tautz. Systematic maintenance of corporate experience repositories. *Computational Intelligence - Special Issue on Maintaining CBR Systems*, 17(2):364–386, May 2001.
[17] M. Nick and R. Feldmann. Guidelines for evaluation and improvement of reuse and experience repository systems through measurement programs. In *Third European Conference on Software Measurements (FESMA-AEMES 2000)*, Madrid, Spain, Oct. 2000.
[18] Organization for Economic Development (OECD). Update on official statistics on internet consumer transactions. http://www.oecd.org/pdf/M00027000/M00027669.pdf, 2001.
[19] K. Racine and Q. Yang. Maintaining unstructured case bases. In *Proceedings of the Second International Conference on Case-Based Reasoning*, pages 553–564, 1997.
[20] R. Rieke. Projects CASENET and SKe - a framework for secure eGovernment. In *Telecities 2002 Winter Conference*, Siena, Italy, Dec. 2002. http://www.comune.siena.it/telecities/program.html.
[21] B. Schneier. *Secrets and Lies: Digital Security in a Networked World*. John Wiley & Sons, 2000.
[22] C. R. Sigrid Gürgens, Peter Ochsenschläger. Role based specification and security analysis of cryptographic protocols using asynchronous product automata. In *DEXA 2002 International Workshop on Trust and Privacy in Digital Business*. IEEE Press, 2002.
[23] C. Tautz. *Customizing Software Engineering Experience Management Systems to Organizational Needs*. PhD thesis, University of Kaiserslautern, Germany, 2001.
[24] The Council for Excellence in Goverment. Poll "eGoverment - The Next American Revolution". http://www.excelgov.org/, Sept. 2000.
[25] M. A. Wimmer and B. von Bredow. Sicherheitskonzepte für e-Government. Technische vs. ganzheitliche Ansätze. *DuD - Datenschutz und Datensicherheit*, (26):536–541, Sept. 2002.