Simple Mode: Addressing Knowledge Engineering Complexity in a Privacy Expert System

Siani Pearson

HP Laboratories
HPL-2010-75

Keyword(s):

privacy, decision support, usability, knowledge engineering

Abstract:

This paper shows how a combination of usability and heuristics can be used to reduce complexity for privacy experts who create and maintain the knowledge base of a decision support system. This system helps people take privacy into account during decision making without being overwhelmed by the complexity of different national and sector-specific legislation.

External Posting Date: July 6, 2010 [Fulltext] Approved for External Publication
Internal Posting Date: July 6, 2010 [Fulltext]
Published as: Siani Pearson, "Addressing Complexity in a Privacy Expert System", E. Hullermeier, R. Kruse, and F. Hoffmann (Eds.): IPMU 2010, Part II, CCIS 81, pp. 612-621, Springer-Verlag Berlin Heidelberg, 2010.

Simple Mode: Addressing Knowledge Engineering Complexity in a Privacy Expert System

Siani Pearson

HP Labs, Long Down Avenue, Stoke Gifford, Bristol, BS34 8QZ, UK
Siani.Pearson@hp.com

Abstract. This paper shows how a combination of usability and heuristics can be used to reduce complexity for privacy experts who create and maintain the knowledge base of a decision support system. This system helps people take privacy into account during decision making without being overwhelmed by the complexity of different national and sector-specific legislation.

Keywords: privacy, decision support, usability, knowledge engineering

1 Introduction

Privacy management for multinational companies is challenging due to the complex web of legal requirements and movement of data and business operations to cost-effective locations. Privacy requirements need to be addressed by dispersed teams, within the context of a variety of business processes, in a global context, and increasingly by people with little knowledge of privacy who have to handle personal data as part of their job. Organisational privacy rulebooks often run into hundreds of pages, and so it is not practical to expect employees' to know all of this information. A decision support system (DSS) can help with this problem by addressing the complexity of compliance requirements for end users, and particularly by assisting individuals who are not experts in privacy and security to find out what to do and highlight where they might not be compliant or where their behaviour poses risks.

Such a tool will have a knowledgebase (KB) that needs to be created and updated by experts on an ongoing basis. These experts can potentially be trained in this process, but they will usually be non-IT staff and may find it difficult to handle complex representations. Therefore, there is a need to address the complexity of the KB updating process. In this paper we explain our approach to this issue, which centres on provision of a novel user interface that facilitates arduous knowledge creation and maintenance tasks and reduces the need for training. Our approach is influenced by Alan Kay's maxim that 'Simple things should be simple, complex things should be possible' [1]. In a 'simple mode' for knowledge maintenance, heuristics are used to hide much of the complexity of the underlying representations from end users, and to fill in appropriate settings within the rules to allow creation of a basic functioning rulebase in a non-complex way. In our case study we have focused on privacy but this approach could be used for a number of other domains; however,

privacy is a particularly suitable domain because of the contextual nature of privacy advice.

2 A Privacy Decision Support System

Our DSS is an expert system that captures data about business processes to determine their compliance. The tool supplies individuals who handle data with sufficient information and guidance to ensure that they design their project in compliance.

There are two types of user: end users (who fill in a questionnaire from which a report is generated), and domain experts (who create and maintain the KB). When an end user uses the DSS, they are initially taken through a series of customised questions and, based on their answers, a compliance report is automatically generated. They can use the tool in an educational 'guidance' mode, where their input is not logged, or alternatively in an 'assessment' mode where a report is submitted that scores the project for a list of risk indicators and a record is retained in the database. Where an issue has been identified, guidance is offered online that links into the external information sources and checklists and reminders are provided. In addition to this user perspective, the system provides a domain expert perspective which is a knowledge management interface for KB creation and update.

2.1 The Underlying Rule Representation

The DSS uses a rules engine, for which two types of rules are defined:

1. **question rules:** these automatically generate questions, in order to allow more subtlety in customisation of the questionnaire to the end user's situation
2. **domain rules:** these generate an output report for end users and potentially also for auditing purposes (with associated checklist, indication of risk, etc.)

All these rules have the general form: **when** condition **then** action.

The DSS uses a set of intermediate variables (IMs) to encode meaningful information about the project and drive the questionnaire, e.g. the IM 'project has transborder data flow' indicates that the current context allows transborder data flow.

The questionnaire maps to a tripartite graph structure as illustrated in Figure 1. The left nodes are monotonic expressions involving (question, answer) pairs. The middle partition consists of intermediate nodes that are semantically meaningful IMs. The right set of nodes represents "new" question(s) that will be asked. The question rules map to lines in Figure 1: they have as their conditions a monotonic expression (i.e. Boolean expression built up using & and *v* as logical operators) in IMs and/or (question, answer) pairs and as actions, directives to ask the user some questions or to set some IMs. The domain rules' condition is a Boolean expression in a set of IMs and answers to questions (cf. the conditions column of Figure 1) and they generate as their actions the content of the output report. See [2] for further details.

Further complexity in the rule expressions arises from the following system features, intended to enhance the end user experience:

- Customised help can be provided, by means of using rules where trigger conditions involve (question, answer) pairs and/or IMs and the inference engine is run to determine the appropriate help
- Subsections allow display of questions related to more complex knowledge
- The parameter “breadth first” (BF) or “depth first” (DF) attached to a question controls whether it is added in a ‘drill down’ fashion, i.e. immediately after the question which led to triggering it, or appended at the end of the list of questions
- An IM expression can trigger a set of questions instead of just one within a rule. In that case the order of questions specified by the expert user in the rule is respected when this block of questions is shown

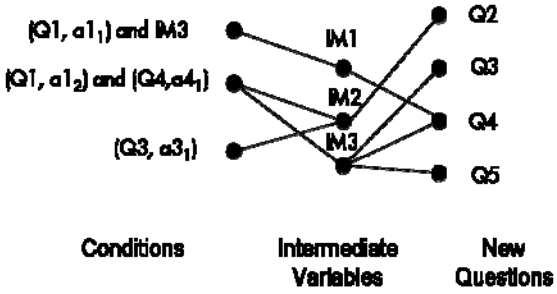

Fig. 1. A representation of the questionnaire using tripartite graphs

Let us consider a simple example of the underlying representation, in DRL format (although the rules can automatically be converted to XML format). Assume that an end user is answering a questionnaire, and that the question “*Is data confined to one country?”* is answered “No”. This (question, answer) pair is added to working memory and as consequence the following question rule is triggered, asserting a new IM “Inv_Transborder_Data_Flow”:

```
rule "IMR21" when QA (id == 48, value == "No") then
insert(new IM("Inv_Transborder_Data_Flow","Yes")); end
```

When the previous IM is asserted to working memory it triggers the following question rule which adds three new questions to the questionnaire:

```
rule "QR17" salience 1000
when IM (name == "Inv_Transborder_Data_Flow", value == "Yes")
then AddToDisplayList_DF(current, currentQuestion, new long[]
{49, 50, 51}); end
```

The initial (question, answer) pair will also generate a new parameter instance: “Data confined to one country” with value "No”. When this parameter instance is added to the working memory of the privacy engine it triggers the following domain rule:

```
rule "Data confined to one country"
when ParameterInstance ( name == "Data confined to one
country" , value == "No"  )
then report.addRule(new RuleFacade().findById(50)); end
```

This rule adds a *Rule* object to the list of rules of the report. The rule will show a yellow flag (to indicate the seriousness of the issue) in the risk indicator "Transborder Data Flow" with the reason: "*Transborder data flow is involved in the project*." More broadly, domain rules can generate as actions other items to be included within the report: a checklist entry which describes what the user should do about the issue raised in this rule; a link to more information.

2.2 Our Implementation: HP Privacy Advisor (HPPA)

HP Privacy Advisor (HPPA) is a DSS of the form described above that supports enterprise accountability: it helps an organisation to ensure privacy concerns are properly and proactively taken into account in decision making in the businesses as well as provide some assurance that this is case [2]. HPPA analyses projects' degree of compliance with HP privacy policy, ethics and global legislation, and integrates privacy risk assessment, education and oversight. Our implementation uses the production rule system Drools [3] for the rules engine and this is run after each question is answered by the user. Since the domain is focused on privacy, we refer to the domain rules as 'privacy rules'.

Several different methods were used for end user testing, and reactions to the tool have been overwhelmingly positive. We have also had validation from privacy experts when learning to use the KB management UIs that the simple mode described in the following section was very helpful, and have undergone a number of iterative improvements to the prototype based upon their suggestions in order to build up a privacy KB. In particular, these experts have entered privacy knowledge using these UIs into the tool that encodes the information from the 300-page HP privacy rulebook.

3 Simple Mode: Simplifying KB Maintenance

With regard to the DSS described in the previous section, the following issue relating to KB maintenance needs to be addressed: how can a non-IT person deal with the complex rule representation and create questionnaires and rules in an easy way?

We found the 'expert mode' screens initially implemented within HPPA for creating and editing question rules and privacy rules too complex for a non-trained person. These screens exposed the representation of the rules in a DRL-type format, as illustrated in Section 2, and also more complex editing that included customised help, tooltips and warnings, question sections, tagging, DF or BF generation of questions, etc. The complexity of this was particularly an issue as the domain experts usually do not have a technical background.

Hence we needed to find a reasonably simple means to update the rules in the KB that would work in the majority of cases and that can be used without the need for training or manipulating the underlying Drools representations. We foresaw two categories of domain experts: those who can carry out simple KB changes and build new questionnaires and those able to fine-tune the rules in the system. We designed a 'simple mode' for the former that could also be used by the latter. Our approach was

to combine intuitive UIs with heuristics that hide the underlying complexity, as follows.

3.1 Usability Aspects

For the question rules, we designed a closer link between authoring and the finished questionnaire. The authoring environment resembles the questionnaire in layout, and the authoring vocabulary is closer to the vocabulary of use (i.e. not rules and variables but questions and answers): if you answer A then you are asked the follow up question B, and so on. The previewing of question sequences allows users to quickly switch from previewing a question in a sequence to editing that same question. The input screens for the privacy rules were also simplified.

We decided to restrict the interface for ‘simple mode’ to a small set of possible constructs. We actively fought against 'feature-creep’, taking our goal for this mode as an interface that is restrictive. We had to balance restrictions against increased ease of learning and use: users can always enlist help or undertake training to achieve more complex goals, using the expert screens.

3.2 Heuristics

Analysis of our KB helped focus attention on the ‘simple’ tasks which make up the majority of the rules which are actually likely to be written by privacy experts, e.g.:

- Most questions had answers ‘yes’, ‘no’ and ‘do not know’
- Most question-setting IMs had a trigger condition of the form: “When QA(id==ID, value==Value)”
- Most privacy rules had a trigger condition of the form: “When Parameter is Value”

The simplified UIs focus on making it easy to do these tasks; heuristics are used to hide the complexity of the underlying representation. In general they enable translation of the user requirements coming from the UIs into the machine readable formats of the rules discussed in Section 2. Thereby, Drools representations and IMs are not exposed to the simple user, and the corresponding ‘simple’ rules are built up by the system. There is no differentiation in the KB about rules derived from expert or simple mode, and this is instead derived from analysing those formats that can be manipulated by simple mode: if a privacy rule is created from the expert screens and has a complex trigger condition then the user is directed to switch to the expert mode to view and edit the rule; otherwise it may be edited within simple mode.

Examples of heuristics used include the following:

- governing whether the rules generated are BF or DF. For instance, when building the questionnaire, users can add follow-on questions; the BF and DF rules are separated by using the section information stored within the follow-up question itself. Questions in the same section as the parent question are made to be DF and questions from different sections are paired and saved as BF mode question rules.
- analysing whether rules need to be combined in order to express more than one follow-up question being generated

- generating IMs when questions are created in simple mode in order to automatically create the corresponding question rules
  In addition, the following mechanisms are used:
- inheriting tags (used in order to identify subject domains) from higher levels in the questionnaire hierarchy (although the user can override this)

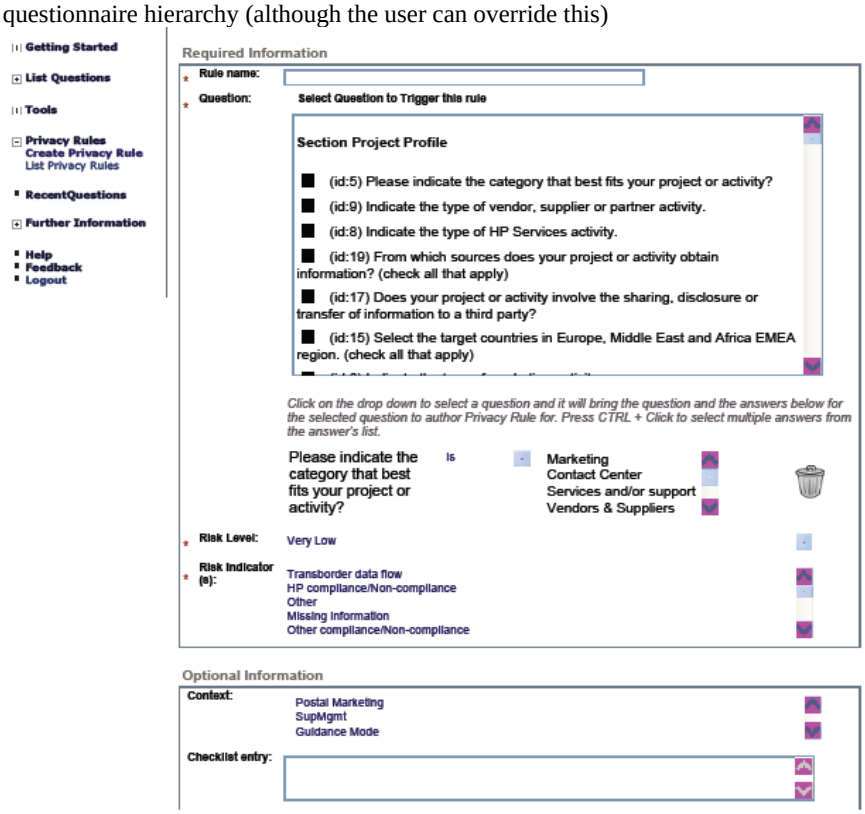

Fig. 2. Create Privacy Rule in Simple Mode

- maintaining a list of 'incomplete' nodes within the questionnaire 'tree' that the user should return to in order to complete the questionnaire. For example, if all answers to a question have follow-up questions defined or are marked as complete then the question is removed from the 'incomplete' list
- preventing the user defining recursive chains when building up the questionnaire by checking there is no duplication of questions in each path

Despite the use of such mechanisms, we found that there are some aspects of the underlying system whose complexity is difficult to avoid and where the resulting solution could still be confusing to the user, notably:

- There is a need to distinguish between 'guidance' and 'assessment' mode mappings. Our solution was to categorise the rules into three modes that can be

selected by users: ‘guidance’, ‘assessment’ or ‘both’: this obviates the need to find out the intersection or union of the mappings that exist in both modes. The active mode can be selected in the ‘list questions’ screen with the default selection as ‘assessment’. Hence, if the context is set as ‘assessment’ then all the filters are done for that mode, so all question rules are checked for the mode selected before modifying them and the rules for ‘guidance’ or ‘both’ are not changed.

- Certain edits could cause major ramifications for other rules: for example, if a user edits a question (for example, amending answer text) that has follow-up questions defined then it is difficult to predict whether or not to keep or break the corresponding links, and to what extent to highlight the effect on the privacy rules that might be triggered – directly or indirectly – by the original question but not the amended version. It is difficult to come up with a heuristic to decide accurately whether the associated rules should be amended or deleted, and so the user should be involved in this decision. Our solution to this was to show a notification to the user in simple mode that this affects the associated rules if they want to make this change, before they make it. They then have the choice whether this edit is automatically propagated throughout the rules, or whether to check the consequences via the expert mode, where the detailed ramifications on the other rules are displayed.

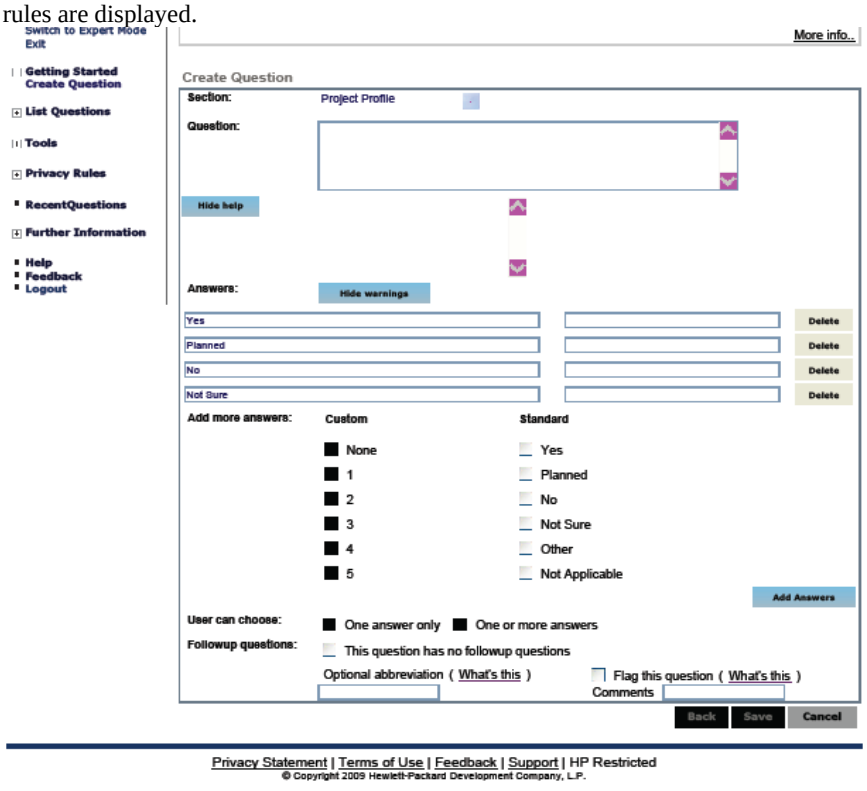

Fig. 3. Create Questionnaire Rule in Simple Mode

As discussed in Section 4, the translation from privacy laws to human-readable policies to machine-readable policies cannot be an exact one. We assume that the privacy expert is able to express in a semi-formalised manner corporate privacy policies or similar prescriptive rules that can be input directly via the UIs and then we automatically encode these into the system rules. Corporate privacy policies would already be close to a suitable form: for example, as illustrated in Figure 2, the 'simple mode' input required to create a privacy rule is: a rule description; the question and answer(s) that triggers the output; what the output is (i.e. the risk level, risk indicator and optional information). A similar approach is taken for screens that allow editing.

Figure 3 illustrates how a simplified approach can be provided to enable generation of question rules. Additional screens allow creation and linkage of follow-on questions, editing question rules, listing questions (and subsets of the KB e.g. tagged questions), simple mode help and previewing the questionnaire (in the sense of stepping through paths of the questionnaire to try it out); for space reasons we are unable to display these UIs in this paper. The system can also highlight parts of the questionnaire that are unfinished, so that the user can complete these. Figure 4 shows how the privacy rules KB may be viewed in an intuitive form.

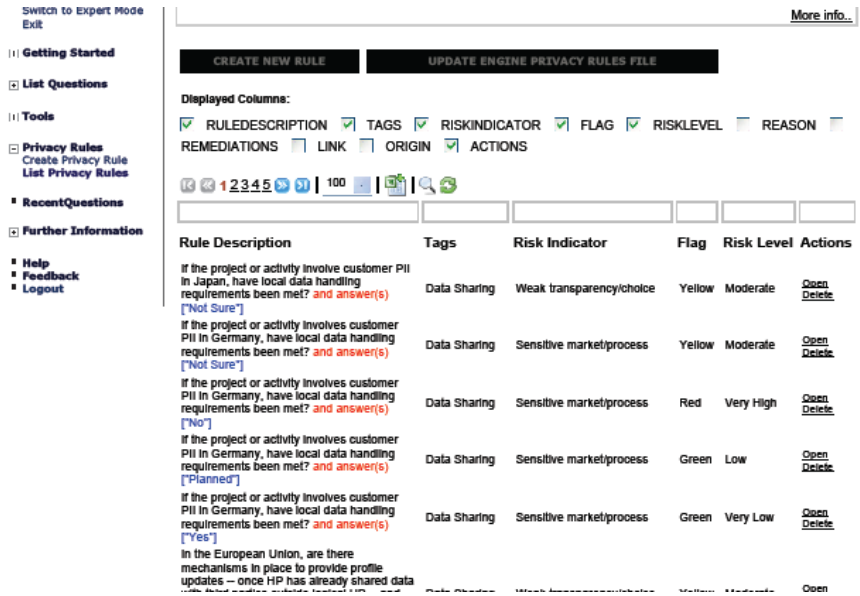

Fig. 4. List Privacy Rules in Simple Mode

A number of open issues remain and we are working to refine our solutions. For example, all kinds of questions in natural language are allowed. Therefore, the system cannot automatically identify duplication of questions that are semantically equivalent but syntactically different. We do solve a restricted from of this problem by

requesting the user to check a box when editing questions to indicate whether or not the new content is semantically equivalent to the old content, and hence enabling us to maintain the relationships between the corresponding rules in the former case.

4 Related Work

Policy specification, modelling and verification tools include EPAL [4], OASIS XACML [5], W3C P3P [6] and Ponder [7]. These policies, however, are at a different level to the ones we are dealing with in this paper, as for example they deal with operational policies, access control constraints, etc. and not a representation of country or context-specific privacy requirements. In addition they are targeted towards machine execution and the question of intermediate, human-actionable representation of policies has so far not been paid attention to in the policy research community. Related technologies in the Sparcle [8] and REALM projects [9] do not produce output useful for humans. OASIS LegalXML [10] has worked on creation and management of contract documents and terms, but this converts legal documents into an XML format that is too long to be human readable and not at the right level for the representation we need in our system. Breaux and Antón [11] have also carried out some work on how to extract privacy rules and regulations from natural language text. This type of work has a different focus then ours but could potentially be complementary in helping to populate the KB more easily. Translation of legislation/regulation to machine readable policies has proven very difficult, although there are some examples of how translations of principles into machine readable policies can be done, e.g. PISA project [12], P3P [6] and PRIME project [13].

The tool we have built is a type of expert system, as problem expertise is encoded in the data structures rather than the programs and the inference rules are authored by a domain expert. Techniques for building expert systems are well known [14]. A key advantage of this approach is that it is easier for the expert to understand or modify statements relating to their expertise. Our system can also be viewed as a DSS. Many different DSS generator products are available, including [15,16]. All use decision trees or decision tables which is not suitable for our use as global privacy knowledge is too complex to be easily captured (and elicited) via decision trees. Rule based systems and expert systems allow more flexibility for knowledge representation but their use demands great care: our rule representation is designed to have some important key properties such as completeness (for further details about the formal properties of our system, see [2]). There has also been some work on dynamic question generation in the expert system community [17,18] but their concerns and methods are very different.

Our research differs from preceding research in that we define an intermediate layer of policy representation that reflects privacy principles linked into an interpretation of legislation and corporate policies and that is human-actionable and allows triggering of customised privacy advice. The focus of this paper is novel use of a combination of heuristics and usability techniques to hide underlying system complexity from domain experts who create and maintain the KB.

5 Status and Conclusions

HPPA has transferred from HP Labs into a production environment and is being rolled out to HP employees in 2010. HPPA tackles complexity of international regulations, helping both expert and non-expert end users with identifying and addressing privacy requirements for a given context. Although our focus has been on privacy, this approach is applicable in a broader sense as it can also apply to other compliance areas, such as data retention, security, and export regulation.

In order to help privacy experts address the complexity of updating KBs in an expert system, a simple mode UI was implemented in HPPA in addition to expert mode screens. Both have been subject to recursive testing and improvement. Some of the features developed for simple mode have subsequently been incorporated into expert mode, in order to improve the usability of those screens. We have also implemented quarantine of rules built up in the simple mode, so that these can be run in test mode before being incorporated into the KB.

Acknowledgments. Lon Barfield advised as to the usability aspects of this problem. The implementation was carried out and benefitted from suggestions by Venkat Dandamundi and Pranav Sharma. Further refinements were made in response to user testing and HP Privacy Office using simple mode screens to input information into HPPA. HPPA is a collaboration between an extended team.

References

1. Leuf, B., Cunningham, W.: The Wiki Way: Quick Collaboration on the Web. Addison-Wesley (2001)
2. Pearson, S., Rao, P., Sander, T., Parry, A., Paull, A., Patruni, S., Dandamudi-Ratnakar, V., Sharma, P.: Scalable, Accountable Privacy Management for Large Organizations, INSPEC'09, IEEE (2009).
3. Drools, http://jboss.org/drools/.
4. IBM: The Enterprise Privacy Authorization Language (EPAL), EPAL specification, v1.2, http://www.zurich.ibm.com/security/enterprise-privacy/epal/ (2004)
5. OASIS: eXtensible Access Control Markup Language (XACML), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
6. Cranor, L.: Web Privacy with P3P, O'Reilly & Associates (2002)
7. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The Ponder Policy Specification Language, 2001, http://www-dse.doc.ic.ac.uk/research/policies/index.shtml
8. IBM: Sparcle project, http://domino.research.ibm.com/comm/research_projects.nsf/pages/sparcle.index.html
9. IBM: REALM project, http://www.zurich.ibm.com/security/publications/2006/REALM-at-IRIS2006-20060217.pdf
10. OASIS: eContracts Specification v1.0, www.oasis-open.org/apps/org/workgroup/legalxml-econtracts (2007)
11. Travis, D., Breaux, T.D., Antón, A.I.: Analyzing Regulatory Rules for Privacy and Security Requirements. IEEE Transactions on Software Engineering, 34(1), pp. 5-20 (2008)
12. Kenny, S., Borking, J.: The Value of Privacy Engineering, JILT (2002)
13. Privacy and Identity Management for Europe. http://www.prime-project.org.eu (2008)

14. Russel, S, Norvig, P.: Artificial Intelligence – A Modern Approach, Prentice Hall (2003)
15.Dicodess: Open Source Model-Driven DSS Generator, http://dicodess.sourceforge.net
16.XpertRule: Knowledge Builder, http://www.xpertrule.com/pages/info_kb.htm
17.J. McGough, J. Mortensen, J. Johnson and S. Fadali, A web-based testing system with dynamic question generation, Proc. Frontiers in Education Conference, Reno, IEEE (2001)
18.J. Bowen and C. Likitvivatanavong, Question-Generation in Constraint-Based Expert Systems, http://ww.4c.ucc.ie