

# A Component Architecture for Dynamically Managing Privacy Constraints in Personalized Web-based Systems

Alfred Kobsa

School of Information and Computer Science  
University of California, Irvine  
<http://www.ics.uci.edu/~kobsa>

**Abstract.** User-adaptive (or “personalized”) systems on the web cater their interaction to each individual user and provide considerable benefits to both users and web vendors. These systems pose privacy problems, however, since they must collect large amounts of personal information to be able to adapt to users, and often do this in a rather inconspicuous manner. The interaction with personalized systems is therefore likely to be affected by users' privacy concerns, and is in many cases also subject to privacy laws and self-regulatory privacy principles. An analysis of nearly 30 international privacy laws revealed that many of them impose severe restrictions not only on the data that may be collected but also on the personalization *methods* that may be employed. For many personalization goals, more than one methods can be used that differ in their data and privacy requirements and their anticipated accuracy and reliability. This paper presents a software architecture that encapsulates the different personalization methods in individual components and, at any point during runtime, ascertains the dynamic selection of the component with the optimal anticipated personalization effects among those that are permissible under the currently prevailing privacy constraints.

## 1 Personalized systems on the web: benefits and methods

User-adaptive (or “personalized”) computer systems take individual characteristics of their current users into account and adapt their behavior accordingly. Such systems have already been deployed in several areas, including education and training (e.g., [1]), online help for complex PC software (e.g., [2, 3]), dynamic information delivery (e.g., [4]), provision of computer access to people with disabilities (e.g., [5, 6]), and to some extent information retrieval. In several of these areas, benefits for users could be empirically demonstrated.

Since about 1998, personalization technology is being deployed to the World Wide Web where it is mostly used for customer relationship management. The aim is to provide value to customers by serving them as individuals and by offering them a unique personal relationship with the business (the terms *micro marketing* and *one-to-one marketing* are being used to describe this business model [7, 8]). Current person-

alization on the web is still relatively simple. Examples include customized content provision (e.g., personalized information on investment opportunities, or personalized news), customized recommendations or advertisements based on past purchase behavior, customized (preferred) pricing, tailored email alerts, and express transactions[9]. Personalization that is likely to be found on the web in the future includes, e.g.,

- product descriptions whose complexity is geared towards the presumed level of user expertise;
- tailored presentations that take users' preferences regarding product presentation and media types (text, graphics, video) into account;
- recommendations that are based on recognized interests and goals of the user; and
- information and recommendations by portable devices that consider the user's location and habits.

A number of studies indicate that users seem to find personalization on the web useful [10, 11], and that they stay longer at personalized websites and visit more pages [12]. Other research demonstrates that personalization also benefits web vendors with respect to the conversion of visitors into buyers [13], “cross-selling” [14], and customer retention and development [15, 16].

Personalized systems utilize numerous techniques for making assumptions about users, such as domain-based inference rules, stereotype techniques, machine learning techniques (e.g content-based filtering, and clique-based or “collaborative” filtering), plan recognition methods, logic-based reasoning, Bayesian inferences, and many more (see [17] for a recent survey). These techniques have different requirements regarding the data that must be available. For instance, most machine learning techniques assume that a large number of raw data (such as a user’s clickstream data) is available and that all learning is performed at one time. Since individual sessions are often too short to deliver sufficient data about a user, these techniques are therefore typically applied to data from several sessions with the user. In contrast, incremental techniques can learn in several steps, taking the new raw data of the current session and the previous learning results into account.

## **2 Privacy problems caused by personalized systems**

Personalized systems generally operate in a data-driven manner: more personalization can be performed the more data about the user is available, and personalization based on more data will also tend to be more accurate and more individualized. User-adaptive systems therefore collect considerable amounts of personal data and “lay them in stock” for possible future usage. Moreover, the collection of information about users is often performed in a relatively inconspicuous manner (such as by watching their web navigation behaviors). Personalized systems are therefore most certainly affected by the privacy concerns that a majority of today’s Internet users articulates, by privacy laws that are in place, and by company and sector privacy policies.

## 2.1 Users' privacy concerns

Numerous consumer surveys have been conducted so far that consistently reveal widespread privacy concerns among today's Internet users.<sup>1</sup> Respondents reported being (very) concerned about, e.g., threats to their privacy when using the Internet (81%–87%), about divulging personal information online (67%–74%), and about being tracked online (54%–77%). They indicated leaving web sites that required registration information (41%) having entered fake registration information (24%–40%), and having refrained from shopping online due to privacy concerns or having bought less (24%–32%). An analysis of results from thirty surveys with a focus on web personalization is given in [18].

Hardly any survey data exists on whether Internet users will agree with the usage of their personal data for personalized interaction. In a poll by an industry advocacy group for web personalization [11], 51% of the respondents indicated to be willing to give out information about themselves in order to receive an “online experience truly personalized for them” (the subjects of this study were however recruited from a “permission-based opt-in list” which may have biased the sample). It seems prudent to assume that the general Internet privacy concerns that were documented by the mentioned consumer surveys also apply to the usage of personal data for web personalization purposes. Caution must be exercised however since users who claim having privacy concerns do not necessarily exhibit a more privacy-minded interaction with web sites, as was demonstrated in experiments by [19].

## 2.2 Privacy laws and self-regulatory privacy principles

Privacy laws protect the data of identified or *identifiable* individuals. For privacy laws to be applicable, it is thus not required that the system actually identifies the user, but only that it is *possible* to identify the user with reasonable efforts based on the data that the system collects. The latter situation often applies to personalized systems. The privacy laws of many countries not only regulate the processing of personal data in the national territory, but also restrict the trans-border flow of personal data, or even extend their scope beyond the national boundaries. Such laws then also affect personalized web sites abroad that serve users in these regulated countries, even when there is no privacy law in place in the jurisdictions in which these sites are located.

We collected nearly 30 international privacy laws and categorized them by criteria that affect the design of personalized systems the most [20]. Categories include registration duties, record-keeping duties, reporting duties, disclosure duties at the website, duty to respect certain user requests, duty to respect user vetoes (“opt out”), duty to ask for user permission (“opt in”), exceptions for very sensitive data, restrictions on data transfer abroad, restrictions on foreign sites collecting data inland, archiving/destruction of personal data, and “other” impacts on personalization. We found that if privacy laws apply to a personalized website, they often not only affect

---

<sup>1</sup> Links to most surveys that are available online can be found at <http://www.privacyexchange.org/iss/surveys/surveys.html>.

the conditions under which personal data may be collected and the rights that data subjects have with respect to their data, but also the *methods* that may be used for processing them. Below is a sample of several legal restrictions that substantially affect the internal operation of personalized hypermedia applications (more constraints will be discussed in the application example).

- *Usage logs must be deleted after each session*, except for billing purposes and certain record-keeping and fraud-related debt recovery purposes [21]. This provision affects, e.g., the above-mentioned machine learning methods that can be employed in a personalized hypermedia system. If learning takes place over several sessions, only incremental methods can be employed since the raw usage data from previous sessions have all to be discarded.
- *Usage logs of different services may not be combined, except for accounting purposes* [21]. This is a severe restriction for so-called central user modeling servers that collect user data from, and make them available to, different user-adaptive applications [22].
- *User profiles are permissible only if pseudonyms are used. Profiles retrievable under pseudonyms may not be combined with data relating to the bearer of the pseudonym* [21]. This clause mandates a Chinese wall between the component that receives data from identifiable users, and the user modeling component which makes generalizations about pseudonymous users and adapts hypermedia pages accordingly.
- *No fully automated individual decisions are allowed* that produce legal effects concerning the data subject or significantly affect him, and which are based solely on automated processing of data intended to evaluate certain personal aspects relating to him, such as his performance at work, creditworthiness, reliability, conduct, etc [23]. This prohibition has impacts on learner-adaptive hypermedia systems for tutoring [24]. E.g., if such systems assign formal grades, there has to be a human in the loop somewhere.
- *Anonymous or pseudonymous access and payment must be offered if technically possible and reasonable* [21, 25].
- *Strong encryption software is regulated in France* [26], which may have impacts on the use of encryption to protect user data in transit when a personalized website or the user is located in France.

In addition to legislative regulations, privacy practices of personalized web sites are also restricted by self-regulatory privacy principles, such as company-specific privacy policies or sector-specific principles (e.g., [27]). These principles can also severely impact the permissibility of personalization methods.

### **3 Privacy management**

#### **3.1 Pseudonymous and identified interaction**

Two principled solutions are possible to cater to privacy requirements in personalized systems. One direction is to allow users to remain anonymous with regard to the

personalized system (and possibly even the whole network infrastructure) whilst enabling it to still link the same user across different sessions, so that it can cater to her individually. In [28, 29], we present a reference model for pseudonymous interaction between users and web-based applications in which full personalization can nevertheless take place.<sup>2</sup>

Pseudonymous interaction seems to be appreciated by users, even though only a single user poll addressed this question explicitly so far [30]. One can expect that anonymity will encourage users to be more open when interacting with a personalized system, thus facilitating and improving the adaptation to this user. The fact that in most cases privacy laws do not apply any more when interaction is anonymous also relieves the application provider from restrictions and duties imposed by such laws. Finally, anonymous and pseudonymous interaction are sometimes even legally mandated if they can be realized with reasonable effort [21, 25].

Anonymous and pseudonymous interaction also has several drawbacks though: it requires an elaborate anonymity infrastructure, it is currently difficult to preserve when payments, physical goods and non-electronic services are being exchanged, and it harbors the risk of misuse. Anonymous personalization is also restricted to electronic channels only. Pseudonymous data cannot be used for cross-channel communication (sending a brochure to a web customer by mail) and cross-channel recognition (recognizing a web customer in a brick and mortar store). These drawbacks become increasingly important since the number of web-only vendors is constantly shrinking.

In the second principled approach to rejoining personalization and privacy, the user would not remain anonymous. Privacy issues are taken into account by respecting privacy laws, self-regulatory privacy principles, and/or users' privacy preferences. This paper deals exclusively with this second approach. It is specifically concerned with architectural issues of privacy management in personalized systems, i.e. software architectures and processes that allow a personalized system to dynamically cater to user's privacy wishes and to regulatory constraints.

### 3.2 Current work on privacy management

Current work in privacy management is mostly concerned with the specification of privacy constraints for data, relating these constraints to software and business processes, and enforcing privacy constraints automatically. [31] introduces privacy meta-data tables which indicate the external recipients and the retention period, for each usage purpose and for each piece of information (attribute) collected for that purpose. A second meta-table specifies access permissions. Processes like the Privacy Constraint Validator, the Attribute Access Control and the Data Retention Manager check the compliance with privacy preferences and privacy policies.

IBM's Enterprise Privacy Architecture [32, 33] maps customer preferences and data onto business processes, privacy rules, technology and the enterprise architecture as a

---

<sup>2</sup> This model also protects the anonymity of the central user modeling server that contains the user's data since knowledge about its location may reveal the identity of the user, e.g. when it is hosted in the user's local network.

whole, and thereby provides a mechanism for analyzing business processes in a privacy context. A “technical reference model” helps guarantee privacy at the transactional level. This model relies on object, data and rules models to build applications that support and enhance privacy and collectively determine what privacy-relevant data is collected and how it must be handled. An authorization director evaluates the given policies and decides whether or not access requests to data sources are granted. [34] focuses on the formulation of enterprise-independent privacy policies in the E-P3P Privacy Policy Language to express and enforce access restrictions to personal data in legacy systems. In a similar vein, [35] study the more expressive logic-based Authorization Specification Language.

[36] presents a formal security model based on state machines, to enforce legal privacy requirements (such as purpose binding or necessity of data processing). The model is based on the integrity concepts of well-formed transactions and separation of duty.

This work complements existing approaches in that it focuses on ways in which a personalized system can dynamically adjust to the currently prevailing privacy constraints. Numerous stipulations in privacy laws, and most likely also user privacy concerns, influence personalization methods in very different ways. A global pre-formulated policy for the selection of personalization methods under each different combination of impact factors does not seem feasible. Instead, a set of personalization methods must be dynamically selected at runtime, considering the current privacy constraints within the general goal of maximizing personalization benefits. In the remainder of this paper, we will discuss an architecture that utilizes functionally related software components for this purpose.

## 4 Redundant-component architectures

### 4.1 Overview

Component architectures have been widely advocated as a means for flexibly assembling software objects both at design time and at run time (e.g., [37, 38]). A *redundant component array* (RAIC) [39-41] is a group of similar or identical components. It uses the services from one or more components inside the group to provide services to applications. Applications connect to a RAIC and use it as a single component. They typically do not know the individual components that underlie a RAIC. Component membership in a RAIC can be *static* or *dynamic*. Components in a static RAIC are explicitly assigned at design time whereas components in a dynamic RAIC may still be incorporated during run-time.

Depending on the types and relations of components in a RAIC, it can be used for many different purposes such as providing higher reliability, better performance, or greater flexibility than what could be achieved by a single component alone. In this paper, we will restrict ourselves to those aspects of RAICs that are relevant for personalization purposes.

Three major types of relations govern the relationship between components in RAICs:

**Interface relations:** Interfaces determine the way in which applications interact with components. Components  $A$  and  $B$  have *inclusionary* interfaces (abbreviated  $A \sqsubseteq_I B$ ) if and only if every possible call to each function in the interface that  $A$  implements can be converted to a call to a corresponding function in  $B$ 's interface without loss of information. Stricter kinds of interface relations are *identical* and *equivalent*; other types are *similar* and *incomparable* (see [39] for their definitions).

**Domain relations:** The domain of a component refers to the scope in which it can provide service, i.e. the range of its input data. Two components  $A$  and  $B$  are said to have *inclusionary* domains (abbreviated  $A \sqsubseteq_D B$ ) if and only if  $A$ 's domain is a subset of  $B$ 's domain, i.e. each input in  $A$ 's valid input domain is also in  $B$ 's valid input domain. A stricter kind of domain relation is *identical*; other types are *exclusionary* and *incomparable*.

**Functional relations:** Functional relations refer to the functionality of components. Two types are relevant for our purposes:

*Similar:* Two components have similar functionalities (abbreviated  $A \approx_F B$ ) if they are designed to perform the same tasks but possibly with different requirements (e.g., with different accuracy).

*Inclusionary:* Two components  $A$  and  $B$  have inclusionary functionality (abbreviated  $A \sqsubseteq_F B$ ) if and only if the functionalities of component  $A$  form a subset of those of  $B$  (i.e., if every possible task that  $A$  performs can be carried out by  $B$ , possibly with different accuracy).

Inclusionary functionality is obviously more general than functional similarity. Even stricter relations are functional *equivalence* and *identity*, but they are not relevant for our purposes.

Relations between two components in RAICs can be “manually” identified, by analyzing their interfaces, service domains and functionality. [39] discusses methods to also determine the relations between components automatically. For certain types of analysis, type information is required that is currently not generally available (such as the “reflection” information on the .NET platform for the analysis of interface relations, or a typology of functionality for the analysis of the functional relations).

A *RAIC controller*, among other things, determines which component(s) inside the RAIC should deliver the services that are offered by the RAIC. For our purposes, the decision is based on a partial order  $<$  between components, the so-called *activation preference*.  $A < B$  denotes that services to the application should be delivered by  $B$  rather than  $A$  if both are in principle eligible. The relationship between two components in this order can be determined empirically (“when  $A$  and  $B$  could both deliver a certain service, which of them should be preferred?”). In many domains (e.g., personalization), an approximation of the activation preference can also be computed based on component relations.

## 4.2 Redundant personalization components

The central tenets of this work are:

1. A personalized system must dynamically cater to changing privacy concerns of users during runtime, and to privacy laws that are in effect in the jurisdictions of both the user and the data processor (and possibly other jurisdictions as well if part of the personal data is located or processed therein).
2. User preferences and privacy laws may have an impact on both the usable data and the permissible methods.
3. A personalized system can dynamically cater to these (changing) requirements when it is designed in a RAIC-like architecture, where RAICs contain functionally inclusionary or at least similar components. At any given time, the services of the RAIC are delivered by that component that is both ranked highest in the activation preference order and meets all current privacy requirements. If a component cannot operate any more due to a change in the privacy requirements, a substitute component is selected based on the activation preference order.

The activation preference order depends on the application domain. For personalization purposes, an approximation of this relation can be constructed based on component relations, using the following rules:

(T 1) If  $A \sqsupseteq_F B$  and  $A \sqsupseteq_I B$ , then  $A < B$

(i.e., if  $B$ 's functionality and interface includes  $A$ 's functionality and interface, then  $B$  should be preferred)

(T 2) If  $A \approx_F B$  and  $A \sqsupseteq_I B$  and  $A \sqsupseteq_D B$ , then  $A < B$

(i.e., if  $A$  and  $B$  are functionally at least similar, and  $B$  uses more data than  $A$  and includes  $A$ 's interface, then  $B$  should be preferred since it will probably deliver higher-quality results).

## 4.3 An example in a personalized recommender domain

We will illustrate our approach using the example of a web store that gives personalized purchase recommendations to web visitors by predicting items in which the user is presumably interested. The service 'predicting the user's interest' is delivered by a RAIC that contains five different components. These components generate predictions based on different data, and use different methods for this purpose (see [17] for a more comprehensive survey of interest prediction methods):

Component A: makes predictions based on the user's demographic data (age, gender, profession, ZIP), by drawing conclusions based on market segmentation data;

Component B: makes predictions based on the user's page visits (during the current session only), using "quick" one-time machine learning methods;



Component C: makes predictions based on the user's demographic data and page visits (in the current session only), using a combination of the methods in A and B;

Component D: makes predictions based on the user's page visits during several sessions, using incremental machine learning methods (the user trace is thereby stored between sessions)

Component E: makes predictions based on the user's demographic data and her page visits during several sessions, using a combination of the methods in A and D (the user trace is again stored between sessions).

Through domain analysis at design time, we can determine that

$$(1) A \approx_{\mathcal{F}} B \approx_{\mathcal{F}} C \approx_{\mathcal{F}} D \approx_{\mathcal{F}} E$$

In the future, (1) may be inferable by meta-descriptions included in every personalization component that locates the component in a function taxonomy.

The following additional relations can all be automatically determined at design time, possibly with the help of limited meta information [39]:

$$(2) A \sqsubseteq_I C, B \sqsubseteq_I C, B \sqsubseteq_I D, C \sqsubseteq_I E, A \sqsubseteq_I E, D \sqsubseteq_I E$$

$$(3) A \sqsubseteq_D C, B \sqsubseteq_D C, B \sqsubseteq_D D, C \sqsubseteq_D E, A \sqsubseteq_D E, D \sqsubseteq_D E$$

With (T 2) we can now conclude that

$$(4) A < C, B < C < E, D < E$$

Based on this partial activation preference order, the RAIC controller can determine that E should be used with highest priority, and that C or D should be used as substitutes if E does not meet the current privacy constraints. (4) is however only an automatically generated approximation of  $<$ . Additional preferences may be entered based on domain knowledge (such as that  $A < B$ , with the – empirically unproven – rationale that interest predictions based on users' individual web navigation outperform predictions based on users' demographic profiles).

Components  $A \sqsubseteq E$  have numerous prerequisites for their operation, which may change during runtime and therefore have to be continuously verified:

1. *Availability of data*: A will not be able to operate if the user did not provide the necessary demographic data. B and D will not be able to operate during the first few interactions with a new user.
2. *Privacy laws*: In many jurisdictions (e.g., in all EU member states [23]), components  $A \sqsubseteq E$  may only operate if the user has unambiguously given her consent to the processing of her personal data for the purpose of personalized interaction. Even when such a general consent was given, the user still has the right to specifically opt out of B if the web store is located in Germany [21, 25]. C and E are illegal for German web stores without the user's consent since use profiles may only be constructed pseudonymously and may not be combined with data of the bearer of the pseudonym. D is illegal in Germany without the user's consent since use data

of online services may only be stored beyond a user session for billing purposes and certain record-keeping and fraud-related fee recovery purposes.

3. *Self-regulatory privacy principles.* If the web store is a signatory of the U.S. Network Advertisers Initiative, C and E may not operate unless the user has consented to the merger of non-personally identifiable use data and demographic data (if the latter is personally identifiable [27]).
4. *Users' individual privacy preferences:* **RAIC** should not operate if the user communicates to the web store that he “does not like being watched while browsing the web store”.

A considerable amount of work already exists on how to communicate users' privacy preferences [42], how to formalize textual privacy policies [34, 35, 43], and how to compare policy requirements with permissions that were given by the users [31, 33]. A decision about which of the components **RAIC** is allowed to operate at a given time can be made using any of these methods, and we will therefore not deal with this issue here. The RAIC can use (4) at any given time to determine which of the permissible personalization components *should* operate since this component provides optimal personalization under the given privacy constraints.

## 5 Conclusion

While personalization on the web is demonstrably beneficial for both web users and web vendors, privacy issues pose a severe obstacle to its broad dissemination. If the user's identity is known to the system or if the user is identifiable, personalization is subject to privacy laws, self-regulatory privacy principles and individual user concerns. These constraints not only affect the kinds of data that may be used for personalization purposes, but also the admissibility of the numerous personalization methods that have been developed to date.

This paper discussed a software architecture in which personalization methods are individually embodied in software components, and where components with similar functionality but different data and privacy requirements are placed into groups (the so-called RAICs) that offer services to applications collectively. Applications that utilize services of a RAIC are unaware of its internal structure, and of the component that currently provides these services. An activation preference order instructs a RAIC controller which component should preferably deliver these services if more than one component meet the current privacy requirements. This architecture allows for a flexible dynamic adjustment of personalization methods to the currently prevailing privacy demands without burdening the application with privacy management tasks.

## References

1. Corbett, A., McLaughlin, M., and Scarpinato, K. C.: Modeling Student Knowledge: Cognitive Tutors in High School and College. *User Modeling and User-Adapted Interaction* 10, (2000) 81-108.
2. Strachan, L., Anderson, J., Sneesby, M., and Evans, M.: Minimalist User Modelling in a Complex Commercial Software System. *User Modeling and User-Adapted Interaction* 10, (2000) 109-146.
3. Linton, F. and Schaefer, H.-P.: Recommender Systems for Learning: Building User and Expert Models through Long-Term Observation of Application Use. *User Modeling and User-Adapted Interaction* 10, (2000) 181-208.
4. Billsus, D. and Pazzani, M. J.: User Modeling for Adaptive News Access. *User Modeling and User-Adapted Interaction* 10, (2000) 147-180.
5. Keates, S., Langdon, P., Clarkson, P., and Robinson, P.: User Models and User Physical Capability. *User Modeling and User-Adapted Interaction* 12, (2002) 139-169
6. Kobsa, A.: Adapting Web Information to Disabled and Elderly Users (invited talk). *WebNet-99*, Honolulu, HI (1999), <http://www.ics.uci.edu/~kobsa/papers/1999-webnet99-kobsa.pdf>.
7. Peppers, D. and Rogers, M.: *The One to One Future: Building Relationships One Customer at a Time*. New York, N.Y.: Currency Doubleday (1993).
8. Peppers, D. and Rogers, M.: *Enterprise One to One: Tools for Competing in the Interactive Age*. New York, N.Y.: Currency Doubleday (1997).
9. Forrester Research: *The Privacy Best Practise*. Cambridge, MA Sept. (1999).
10. Hof, R., Green, H., and Himmelstein, L.: Now it's YOUR WEB. *Business Week* Oct. 5 (1998) 68-75.
11. Personalization & Privacy Survey. Personalization Consortium (2000), <http://www.personalization.org/SurveyResults.pdf>
12. Thompson, M.: Registered Visitors Are a Portal's Best Friend. *The Industry Standard*, June 7, 1999, <http://www.thestandard.net>
13. More Concentrated than the Leading Brand. *ICONOCAST* (1999), <http://www.iconocast.com/icono-archive/icono.102199.html>
14. Recommender Systems in E-Commerce. (2000), <http://www.cs.umn.edu/Research/GroupLens/slides-2.pdf>
15. Cooperstein, D., Delhagen, K., Aber, A., and Levin, K.: *Making Net Shoppers Loyal*. Forrester Research, Cambridge, MA June (1999).
16. Peppers, D., Rogers, M., and Dorf, B.: *The One to One Fieldbook*. New York, NY: Currency Doubleday (1999).
17. Kobsa, A., Koenemann, J., and Pohl, W.: Personalized Hypermedia Presentation Techniques for Improving Customer Relationships. *The Knowledge Engineering Review* 16 (2001) 111-155, <http://www.ics.uci.edu/~kobsa/papers/2001-KER-kobsa.pdf>.
18. Teltzrow, M. and Kobsa, A.: Impacts of User Privacy Preferences on Personalized Systems - a Comparative Study. *CHI-2003 Workshop "Designing Personalized User Experiences for eCommerce: Theory, Methods, and Research"*, Fort Lauderdale, FL (2003), <http://www.ics.uci.edu/~kobsa/papers/2003-CHI-teltzrow-kobsa.pdf>.
19. Spiekermann, S., Grossklags, J., and Berendt, B.: E-privacy in 2nd Generation E-Commerce: Privacy Preferences versus Actual Behavior. *EC'01: Third ACM Conference on Electronic Commerce*, Tampa, FL (2001) 38-47, <http://doi.acm.org/10.1145/501158.501163>.

20. A Collection and Systematization of International Privacy Laws, with Special Consideration of Internationally Operating Personalized Websites. (2002), <http://www.ics.uci.edu/~kobsa/privacy>
21. Teleservices Data Protection Law (Article 3 of the Law on the Legal Requirements for Electronic Business Dealings of 14 Dec. 2001). German Federal Law Gazette 1, 3721 (2001), [http://www.iid.de/iukdg/aktuelles/fassung\\_tdg\\_eng.pdf](http://www.iid.de/iukdg/aktuelles/fassung_tdg_eng.pdf)
22. Kobsa, A.: Generic User Modeling Systems. User Modeling and User-Adapted Interaction 11 (2001) 49-63 <http://www.ics.uci.edu/~kobsa/papers/2001-UMUAI-kobsa.pdf>.
23. EU: Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of such Data. Official Journal of the European Communities (1995) 31ff, <http://158.169.50.95:10080/legal/en/dataprot/directiv/directiv.html>.
24. Brusilivsky, P.: Adaptive and Intelligent Technologies for Web-based Education. KI 4, (2000) 19-25, <http://www2.sis.pitt.edu/~peterb/papers/KI-review.html>.
25. EU: Directive 2002/58/EC of the European Parliament and of the Council Concerning the Processing of Personal Data and the Protection of Privacy in the Electronic Communications Sector (2002) <http://register.consilium.eu.int/pdf/en/02/st03/03636en2.pdf>.
26. Décret no 99-200 du 17 mars 1999 définissant les catégories de moyens et de prestations de cryptologie dispensées de toute formalité préalable. Le Journal officiel de la République française, (1999) <http://www.legifrance.gouv.fr/WAspad/UnTexteDeJorf?numjo=PRMX9903477D>.
27. Self-Regulatory Principles for Online Preference Marketing by Network Advisers. Network Advertising Initiative (2000), [http://www.networkadvertising.org/images/NAI\\_Principles.pdf](http://www.networkadvertising.org/images/NAI_Principles.pdf)
28. Schreck, J.: Security and Privacy in User Modeling. Dordrecht, Netherlands: Kluwer Academic Publishers (2003), <http://www.security-and-privacy-in-user-modeling.info>.
29. Kobsa, A. and Schreck, J.: Privacy through Pseudonymity in User-Adaptive Systems. ACM Transactions on Internet Technology 3 (2003), 149-183 <http://www.ics.uci.edu/~kobsa/papers/2003-TOIT-kobsa.pdf>
30. GVU's 10th WWW User Survey. Graphics, Visualization and Usability Lab, Georgia Tech (1998), [http://www.cc.gatech.edu/gvu/user\\_surveys/survey-1998-10/](http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-10/)
31. Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y.: Hippocratic Databases. 28th International Conference on Very Large Databases, Hong Kong, China (2002), <http://www.vldb.org/conf/2002/S05P02.pdf>.
32. Enterprise Privacy Architecture: Securing Returns on E-Business. (2002), <http://www-1.ibm.com/services/files/epaexecbrief.pdf>
33. Karjoth, G., Schunter, M., and Waidner, M.: Privacy-Enabled Services for Enterprises. International Workshop on Trust and Privacy in Digital Business (Trustbus 2002), Aix-en-Provence, France (2002) 483-487.
34. Karjoth, G., Schunter, M., and Waidner, M.: Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data. in 2nd Workshop on Privacy Enhancing Technologies, LNCS. Berlin: Springer-Verlag (2002).
35. Karjoth, G. and Schunter, M.: A Privacy Policy Model for Enterprises. 15th Computer Security Foundations Workshop (CSFW'02), Cape Breton, Nova Scotia, Canada, (2002) 271-281.
36. Fischer-Hübner, S.: IT-Security and Privacy: Design and Use of Privacy-Enhancing Security Mechanisms. LNCS 1958. Heidelberg-Berlin, Germany: Springer (2001).

37. Szyperski, C.: *Component Software: Beyond Object-Oriented Programming*. Reading, MA: Addison-Wesley (1998).
38. Heineman, G. T. and Councill, W. T.: *Component Based Software Engineering: Putting the Pieces Together*. Reading, MA: Addison-Wesley (2001).
39. Liu, C.: *Redundant Arrays of Independent Components*. Irvine, CA: School of Information and Computer Science, University of California (2002).
40. Liu, C. and Richardson, D. J.: *Research Directions in RAICs*. ACM SIGSOFT Software Engineering Notes 27 (2002).
41. Liu, C. and Richardson, D. J.: *The RAIC Architectural Style*. School of Information and Computer Science, University of California, Irvine, CA, Working Paper (2002).
42. A P3P Preference Exchange Language 1.0 (APPEL1.0): W3C Working Draft 15 April (2002), <http://www.w3.org/TR/P3P-preferences>
43. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation 16 April (2002), <http://www.w3.org/TR/P3P/>