PLA-based Runtime Dynamism in Support of Privacy-Enhanced Web Personalization

Yang Wang, Alfred Kobsa, André van der Hoek, Jeffery White
Department of Informatics
Donald Bren School of Information and Computer Sciences
University of California, Irvine
Irvine, CA 92697-3440, U.S.A
{yangwang | kobsa | andre | whitej}@ics.uci.edu

Abstract

Software product line architectures (PLAs) have been widely recognized as a successful approach in industrial software development for improving productivity, software quality and time-to-market. In this paper, we focus on the usage of a PLA for a quite different purpose, namely, handling privacy constraints in web personalization. To provide personalized services such as customized recommendations, a personalized website collects users' personal data, which raises various privacy concerns. We aim at reconciling the benefits of web personalization with privacy constraints that come from users themselves as well as from privacy legislations and regulations that apply to a given user. We propose a dynamic, privacy-enabling personalization infrastructure and conceive it as a PLA. This infrastructure allows for dynamically selecting and instantiating personalization architectures that provide personalized services to each individual user and comply with the prevailing privacy constraints.

1. Introduction and overview

Software product line architectures (PLAs) have gained a lot of momentum in industrial software development because of their benefits in improving productivity, software quality and time-to-market [1-3]. In this paper, we show that PLAs can also be used for a quite different purpose, namely handling privacy constraints in web personalization.

Personalized (or user-adaptive) systems are applications that take individual characteristics of their current users into account and adapt their behavior accordingly. For doing so, they collect considerable amounts of personal data about the user and "lay them in stock" for possible future adaptation. This has proven to be very beneficial for computer users in several application areas including education and training (e.g., [4]), online help for complex PC software (e.g., [5, 6]), dynamic information delivery (e.g., [7]), provision of computer access to people with disabilities (e.g., [8]), and information retrieval systems (e.g., [9]).

Recently, personalization technologies have been successfully introduced on the World Wide Web where they are mostly used for customer relationship management [10, 11]. The single most important way to provide value to customers is to know them and serve them as individuals. Customers need to feel they have a unique personal relationship with the business. Current web personalization examples include customized content (e.g., personalized finance pages or news collections), customized recommendations or advertisements based on past purchase behavior, customized (preferred) pricing, tailored email alerts, and express transactions [12]. A number of studies show that personalization has provided benefits for both online customers and vendors [13, 14].

However, personalization benefits are offset by privacy concerns [12, 15-17]. Web users are not only concerned about their privacy (e.g., about being tracked online), but already counteract (e.g., by leaving websites that require registration information or by entering fake registration information).

Since personalized websites collect personal data, they are also subject to privacy laws and regulations if the respective individuals are in principle identifiable. A review of more than 40 international privacy laws [18] shows that if privacy laws apply to a personalized website, they often not only affect the data that are collected by the website, the way in which the data is transferred and to which party it is transferred, but also the methods that may be used for processing them (and consequently the components that embed such methods). For instance, the German Teleservices Data Protection Act [19] mandates personal data to be erased immediately after each session except for very limited purposes. This provision would affect the use of those machine learning methods where the learning takes place over several sessions.

Our primary focus is on the data processing step of web personalization where numerous personalization methods can be applied to derive additional assumptions about users. It is important to note that for many personalization goals, more than one method can be used, each differing in their data and privacy requirements and often their anticipated accuracy and reliability.

From a personalization point of view, we ask the research question: how can personalized web-based systems maximize their personalization benefits, while being compliant with the privacy constraints that are currently in effect (such as the aforementioned privacy laws, industry and company regulations, and privacy preferences of the current user)? Our vision is to provide personalized privacy management where the personalization process is tailored to each individual user's privacy constraints.

A review of several existing approaches shows that they all fail to provide a flexible, systematic and scalable solution for the enforcement of users' potentially different privacy constraints. Inspired by the idea of treating software as a product line to support software variability from design-time to invocation-time to run-time [20] and several other works in the field of dynamic architecture and run-time architecture evolution [21-25], we propose a dynamic, privacy-enabling personalization infrastructure. Particularly, we propose to leverage the concept of product line architecture to model the variability that exists in the privacy and personalization domain, and to dynamically select architectural instances [24] to tailor the product line architecture to the specific needs of a particular user. The infrastructure, thus, considers the privacy constraints that apply to an individual user and dynamically selects and instantiates a personalization architecture that provides personalized services to this specific user. The result is a flexible approach that not only helps address the complexity of building personalized systems, but also strongly supports their evolution: as new privacy and personalization concerns arise, they can be modularly added to the product line architecture.

The main contributions of this paper are the following:

1. a novel application of PLAs to address a practical and complex socio-technical problem – balancing privacy and personalization; and
2. a PLA-based solution that particularly relies on run-time variability for the dynamic configuration of personalized web-based systems.

In the remainder of this paper, we first discuss several existing approaches to the problem of taking users' potentially differing privacy constraints into account (Section 2). We then present our PLA-based approach (Section 3), a detailed example with our prototype system (Section 4), and finally our conclusions and planned future work (Section 5).

2. Existing approaches

Because specialized infrastructures for building systems that cater to the privacy constraints of individual users do not yet exist, websites that aim at addressing this problem currently have to use simple escape strategies which we list below.

2.1. Pseudonymous interaction

This approach allows users to remain anonymous with regard to the personalized system and the whole network infrastructure, whilst enabling the system to still recognize the same user in different sessions so that it can cater to her individually [26]. At first sight, this seems to be a panacea because in most cases, privacy laws no longer apply when the interaction is anonymous. However, anonymity is currently difficult and/or tedious to preserve when payments, physical goods and non-electronic services are being exchanged. This solution also harbors the risk of misuse and hinders vendors from cross-channel marketing (e.g. sending a products catalog to a web customer by postal mail). Moreover, users may still have additional privacy preferences (e.g., they do not want to be profiled even when it is only done pseudonymously), which this approach does not take into account.

2.2. Largest permissible common subset

Ideally, this approach means that only those personalization methods are used that satisfy all privacy laws and regulations. The Disney website, for instance, observes both the U.S. Children's Online Privacy Protection Act (COPPA) as well as the European Union Directive [27]. This solution is likely to run into problems if more than a very few jurisdictions are involved, since the largest common subset of permissible personalization methods may then become very small.

2.3. Different country/region versions

In this approach, personalized systems have different country versions, each of which uses only those personalization methods that are permitted in the respective country. If some countries have similar privacy laws, their versions can be combined using the above-described largest permissible common subset approach. For example, IBM's German-language pages comply with the privacy laws of Germany, Austria and Switzerland [28], while IBM's U.S. site meets the legal constraints in U.S. As with the largest permissible common subset approach, this approach also has scaling problems as soon as the number of countries/ regions, and hence the number of different versions of the personalized system, increases.

2.4. P3P

The Platform for Privacy Preferences (P3P) [29] enables websites to express their privacy policies in a standard format that can be retrieved automatically and interpreted by user agents. Client-side agents can then inform users about the sites' privacy policies and warn them when those deviate from previously-specified preferences. P3P does not enforce privacy policies nor does it support different policies for different users. By itself, it is therefore not an answer to the need for privacy tailored to different user constraints. However, several proposals for individual negotiation of P3P policies have been made [30, 31]. The results of such negotiations could become the input to our own approach.

2.5. Summary

Pseudonymous interaction bypasses the applicability of privacy laws and regulations at the price of tedious operation, but does not provide sufficient support for dealing with users' own privacy concerns. The largest permissible common subset approach and different country/region versions do not scale up well and cannot address users' individual privacy preferences either. P3P helps websites communicate their privacy policies to users, but does not support different policies for different users. In a nutshell, none of these approaches fulfills our vision of personalized privacy management. They all fail to provide a flexible, systematic and scalable solution for the enforcement of privacy constraints that may differ among users.

3. Our PLA-based approach

Our goal is to achieve maximum personalization benefit while at the same time satisfying the prevailing privacy constraints, at the individual user level. In this section, we describe how we approach the problem in the context of personalized system design, and present our PLA-based personalization infrastructure and its underlying privacy-enabling mechanism in details.

3.1. User Modeling Server

Most personalized systems employ a user modeling system, usually in a client-server fashion, which is then dubbed a User Modeling Server (UMS). A UMS stores and represents user characteristics and behavior, integrates external user-related information, applies user modeling methods to derive additional assumptions about the user, and allows multiple external user-adaptive applications to retrieve user information from the server concurrently [32].

For many personalization goals, more than one user modeling method can be used, each differing in their data and privacy requirements and often their anticipated accuracy and reliability. For example, a personalized website could use incremental machine learning (that discards all raw data after the end of a session) to provide personalization to web visitors from Germany[1], while it can use possibly more accurate one-time machine learning with data from several sessions to provide personalization to web visitors from the U.S. who are not subject to the same privacy constraints.

Since UMSs are the central repositories for personal information in personalized systems and the loci of personal data processing, our solution focuses on using a product line architecture for UMSs, with which we address privacy and personalization issues.

3.2. Our dynamic privacy-enabling personalization infrastructure

Figure 1 shows a high level overview of our privacy-enabling personalization infrastructure. It consists of external user-adaptive applications (e.g., a personalized shopping site), the Selector, and the UMS that includes the Directory Component and a pool of user modeling components (UMCs). External personalized applications can query the UMS for existing user information,

[1] This is not yet a complete solution, though, since the German Teleservices Data Protection Act [19] also mandates that profiling requires the use of pseudonyms or the consent of the user.

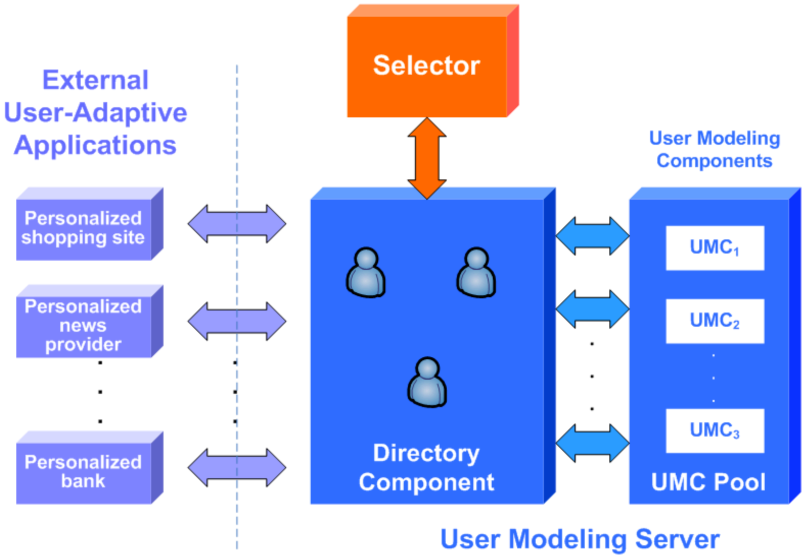


Figure 1. Our dynamic privacy-enabling personalization infrastructure

so as to provide personalized services to their end users, and can supply additional user information to the UMS.

In our solution, we implement the UMS as a PLA. Doing so allows us to provide an infrastructure that solves the problem of handling privacy constraints in web personalization in a generic fashion, to take advantage of commonalities among different needs for privacy and personalization, and to dynamically update different privacy and personalization strategies in a modular fashion, not requiring that the UMS be entirely rebuilt upon each change.

The Directory Component is essentially a repository of user models, each of which stores and represents not only the characteristics, behavior, and inferences about each user, but also their potentially different individual privacy constraints. The UMC Pool contains a set of UMCs, each of which encapsulates one or more user modeling methods (e.g., collaborative filtering [9]) that are utilized in drawing additional inferences about users based on existing user data. Each UMC forms an *optional element* [33] guarded by a Privacy Boolean Expression (PBE; see Section 3.3) in the PLA.

A particular personalization architecture containing only those UMCs that are allowed to operate under a user's prevailing privacy constraints (see Section 3.4) can be selected from the PLA by the Selector, and then instantiated to provide services to the external personalized applications as a UMS for the respective user. Moreover, in order to maximize the benefits of personalization, the Selector can further select the UMCs with the optimal anticipated personalization effects among those that are currently permissible based on a designer-specified preferred order.

It is important to stress that if two or more users have the same set of privacy constraints they will share a single personalization architecture. This reusability is fundamental to making our solution scalable.

3.3. Modeling privacy impacts on UMCs

A Privacy Boolean Expression (PBE) captures whether its associated UMC is allowed to operate under a set of identified privacy concerns. A PBE is a logic combination of Privacy Boolean Variables (PBVs), which are defined during a manual analysis of the impacts of potential privacy concerns on a UMC. If the PBE is resolved to be true, then the associated UMC will be selected in the resulting personalization architecture; otherwise, the UMC will not be included.

For example, assume that a UMC employs one-time machining learning combined with a clustering technique to generate personalized music recommendations for a user. It analyzes both the user's browsing history over several sessions in a personalized online music store and her demographic data such as gender, address and occupation. Table 1 shows the PBVs that have been defined to capture the potential privacy concerns in this scenario, and the resulting PBE.

Table 1. The Privacy Boolean Expression of the example User Modeling Component in Section 3.3, and its constituent Privacy Boolean Variables

PBVs	Name	Corresponding privacy concern
PBV_1	combining_profile	Combining pseudonymous usage data with personally identifiable demographic data
PBV_2	keeping_n_sessions_data	Keeping usage data across sessions
PBV_3	tracking_user	Monitoring user browsing behavior
PBE	combining_profile && keeping_n_sessions_data && tracking_user	

Table 2. The Privacy Constraint Bindings for the example user in Section 3.4

PCBs	Expression	Corresponding privacy constraint
PCB_1	combining_profile = false	German law prohibits combining user profiles retrievable under pseudonyms with data relating to the bearer of the pseudonym.
PCB_2	keeping_n_sessions_data = false	German law mandates personal data to be erased immediately after each session except for very limited purposes.

3.4. Expressing privacy constraints

Privacy constraints that apply to a user can be privacy laws and regulations that are in effect, as well as the user's own personal privacy preferences. Those privacy constraints are expressed in name-value pairs and used as bindings for the Boolean guards associated with each UMC. We call them Privacy Constraint Bindings (PCBs).

For example, we may have a German user who did not mention any personal privacy preferences. Nevertheless, if she is in principle identifiable, the German Teleservices Data Protection Act would apply. Table 2 summarizes the user's privacy constraints and their respective PCBs.

During the evaluation of the PBE in Table 1, each PBV will be bound to its PCB (those in Table 2) for this German user. If such a binding does not exist, we set the default binding to be true. We thus take a "permissive approach": practices not explicitly forbidden by the prevailing privacy constraints (tracking the user in our example) are included. In our example, the UMC will not be selected because its PBE is resolved to false given the PCBs.

3.5. Dynamic selection process

The Selector monitors the start and end of user sessions via bind and unbind operations onto the UMS by the external applications. When the Selector detects the start of a user session, it initiates a *Privacy Context Detection* process that will collect all the active privacy constraints and then generate corresponding PCBs. A similar process will be carried out whenever during a user session the Selector learns about new or changed privacy requirements (which for all practical purposes will stem from user preferences since privacy laws and regulations are unlikely to change during a session).

The PCBs are fed into the Selector that will carry out a *PLA selection process* [24]. Firstly, the PBEs of all UMCs are evaluated based on the given PCBs, to determine whether or not these UMCs may be included in the personalization architecture for the current user session. Secondly, a binary Privacy Constraint Satisfaction (PCS) vector is constructed whose nth element represents whether or not the nth UMC may be used. The Selector checks whether a run-time system instance with such a PCS already exists. If so, the Selector will assign the user session to the existing run-time system instance that has the same PCS. If not, the Selector will perform *PLA Pruning* that automatically removes any disallowed components from the architecture, and then the Selector instantiates a new run-time system instance for the user session. Figure 2 presents the pseudo-code of the above process.

4. A detailed interaction example

In this section we describe a prototype system that we built for proving the concept of our approach.

4.1. The example scenario

Let us assume that UniversalFriends.com is a website that is operated in the USA by a signatory of the U.S. Network Advertisers Initiative (NAI) [34]. The goal of this website is to bridge physical distances between people and to foster world-wide friendships through information technology. It provides personalized services to help customers make friends worldwide. Upon registration, users will be asked to choose a pseudonymous user ID along with a password

```
The Selector monitors the start and end of user sessions:
    On bind (start):
        Privacy Context Detection:
            Collect active privacy constraints;
            Generate variable bindings (PCBs);
        PLA selection, based on PCBs:
            Evaluate Boolean guards (PBEs) for UMCs;
            Construct a new PCS vector V;
        IF there already exists an identical PCS THEN
            Assign the user session to the existing
            run-time system instance, say instance$_i$;
            instance$_i$ . numSessions ++;
        ELSE
            PLA Pruning:
                Prune out UMCs whose PBEs are
                resolved to FALSE;
            Instantiate a new run-time system instance
            for the user session, say instance$_{n+1}$);
            instance$_{n+1}$ . numSessions = 1;
    On unbind (end):
        numSessions$_{current}$ - -;
        IF numSessions$_{current}$ == 0 THEN
            Kill run-time system instance$_{current}$;

If new/changed user privacy preferences are detected, a
similar process starts as on bind.
```

Figure 2. Dynamic selection process

and to disclose some information about themselves (e.g., their hobbies). They will be given some space on the UniversalFriends web server to create their own homepages. The system will recommend a personalized list of likely friends based on a user's characteristics, and will automatically send invitations for pair-wise virtual meetings.

We have three hypothetical users, Alice, Cheng and Bob. Table 3 describes their characteristics.

The UniversalFriends web server relies on our privacy-enabling personalization infrastructure to infer information about users in order to recommend potential friends. Table 4 and Table 5 show the types of input data and the available inference methods, respectively. Table 6 summarizes the usage of data and inference methods for each user modeling component.

For example, UMC$_1$ can recommend people in the same profession cluster. If a user indicates a high interest in a specific topic, UMC$_2$ can infer that she would like to meet people with similar ratings for a topic; alternatively in this case, UMC$_3$ can infer with 95% confidence that she would like to meet people with similar ratings for the topic.

4.2. Interaction with the personalized system

Users can interact with the system as follows:

Table 3. Our hypothetical users

Name	Current location	Personal privacy preference(s)
Alice	Germany	None
Cheng	China	Dislikes being tracked
Bob	USA	None

Table 4. Types of input data

Abbreviation	Type of input data
Demographic	Demographic data such as age, gender, profession, education level
User_supplied	User-supplied data, e.g., a user indicates her levels of interests in different topics
1_Session	UniversalFriends pages that the user visited in the current session
N_Sessions	UniversalFriends pages that the user visited across sessions

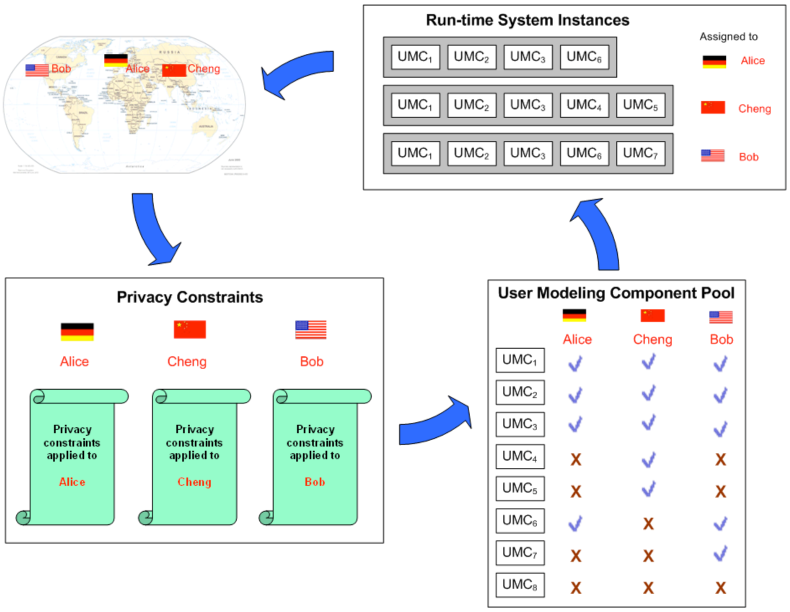
Table 5. Types of inference methods

Abbreviation	Type of inference method
Clustering	Clustering techniques
Rule-based	Rule-based reasoning
Fuzzy	Fuzzy reasoning with uncertainty
Incremental ML	Incremental machine learning
One-time ML	One-time machine learning across several sessions

Table 6. UMCs pool

UMC	Data used	Methods used
UMC$_1$	• Demographic	Clustering
UMC$_2$	• User_supplied	Rule-based
UMC$_3$	• User_supplied	Fuzzy
UMC$_4$	• Demographic • User_supplied	Rule-based
UMC$_5$	• Demographic • User_supplied	Fuzzy
UMC$_6$	• User_supplied • 1_Session	Incremental ML
UMC$_7$	• User_supplied • N_Sessions	One-time ML
UMC$_8$	• Demographic • User_supplied • N_Sessions	One-time ML Fuzzy reasoning

1. Users log into UniversalFriends.com using their registered user names and passwords.
2. The website gathers users' current privacy constraints including those imposed by privacy laws and regulations, and their privacy preferences. Users can specify their privacy


Bob
Alice
Cheng
Run-time System Instances
Assigned to
UMC1 UMC2 UMC3 UMC6 — Alice
UMC1 UMC2 UMC3 UMC4 UMC5 — Cheng
UMC1 UMC2 UMC3 UMC6 UMC7 — Bob
Privacy Constraints
Alice
Cheng
Bob
Privacy constraints applied to Alice
Privacy constraints applied to Cheng
Privacy constraints applied to Bob
User Modeling Component Pool

	Alice	Cheng	Bob
UMC$_1$	✓	✓	✓
UMC$_2$	✓	✓	✓
UMC$_3$	✓	✓	✓
UMC$_4$	X	✓	X
UMC$_5$	X	✓	X
UMC$_6$	✓	X	✓
UMC$_7$	X	X	✓
UMC$_8$	X	X	X

Figure 3. Privacy-enabling personalization process

preferences and change them anytime during the interaction with the personalized system. For instance, if they feel that a specific piece of privacy law or regulation is too strict to get otherwise much better personalization, they can give their consent to certain system actions that are otherwise legally prohibited (e.g., the storage of personal data across sessions).

3. For every user, a summary webpage shows:
   a. their prevailing privacy constraints, and
   b. the selected UMCs used in producing the personalized service, and the excluded UMCs and the reasons for their exclusion (i.e., the specific privacy constraints).

4.3. Privacy-enabling personalization process

The privacy constraints that apply to each of the three individual users and their implications for the UMCs are discussed below (due to limited space, the relevant PCBs and PBEs are not presented here, but they can be defined as described in Sections 3.3 and 3.4):

For Alice, the German Teleservices Data Protection Act applies, with the following consequences:

- UMC$_4$, UMC$_{5,}$ and UMC$_8$ are illegal because the law prohibits combining user profiles retrievable under pseudonyms with data relating to the bearer of the pseudonym.
- UMC$_7$ and UMC$_8$ are illegal because the law mandates personal data to be erased immediately after each session except for very limited purposes.

Therefore, UMC$_4$, UMC$_{5,}$ UMC$_7$ and UMC$_8$ cannot be used for Alice without her explicit consent.

While no privacy law applies to Cheng, she has her own personal privacy preference, such as that she “dislikes being tracked”. Hence UMC$_{6,}$ UMC$_7$ and UMC$_8$ cannot be used because the system may not keep track of the pages she visits on UniversalFriends.com.

For Bob from the United States, UMC$_4$, UMC$_5$ and UMC$_8$ cannot be used according to the NAI self-regulation [34] if he does not consent to merging non-personally identifiable usage data with personally identifiable demographic data.

Figure 3 illustrates the process of selecting and instantiating personalization architectures for each user according to their individual privacy constraints (as we explained in 3.5). Note that, in this case, three different architectural instances are created since each user has different privacy constraints.

4.4. Implementation

The prototype system is currently composed of three basic components: a Context Detector, an Instance Manager, and a light version of ArchStudio [35]. To simplify matters, we did not yet include a Directory Component. Figure 4 gives a high-level overview of the system structure.

The Context Detector is the component that interfaces with a user’s web browser, collecting her privacy constraints and relaying them to the next component, the Instance Manager. The ArchStudio

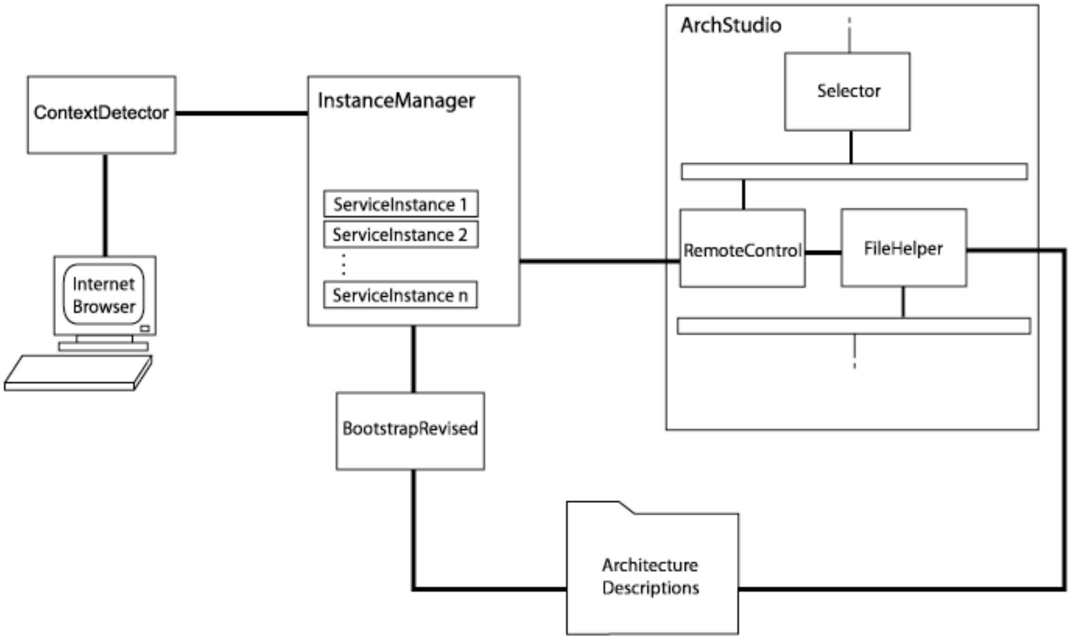


Figure 4. System architecture

component is mainly used for its Selector [24], which generates the architecture descriptions (expressed in xADL 2.0 [36] and selected from an overall PLA description) for the personalization architectures, or "personalized system instances" tailored to each individual user based on their privacy constraints. The Instance Manager is the central core of the system. It responds to the requests of the Context Detector and uses ArchStudio to build the personalized system instances.

All three main components of the system are implemented in Java and communicate via the Java Remote Method Invocation (RMI) framework. Using this method, it is possible for the components to be distributed across more than one machine, but this is currently not the case. The Instance Manager and the RemoteControl subcomponent of ArchStudio extend the remote interface and sign their names to the RMI registry, allowing the Context Detector and ArchStudio to access the Instance Manager directly, as well as allowing the Instance Manager to invoke ArchStudio's Selector functionality.

Minor miscellaneous components of the system include BootstrapRevised (a modified version of the Bootstrapper from the orginal ArchStudio), which the Instance Manager uses to initialize architecture descriptions into running instances. In Figure 4, the stored architecture descriptions produced by ArchStudio are simply represented as a file directory located on the server machine. The web pages produced by the Context Detector are served via Apache Tomcat servlets, which are also able to make requests of the Instance Manager directly once a user's system instance is produced.

When a user first interacts with the system using her web browser, she will be prompted by the Context Detector for her privacy constraints. When submitted, these constraints are transferred to the Instance Manager as a new user request. They are packaged by the Instance Manager and posted to the ArchStudio Component for selection processing. Then a customized architecture is selected and its description saved to a file. The Instance Manager, thereupon, receives a request to instantiate the newly completed architecture. It first analyzes the new architecture to construct a PCS Vector describing which UMCs are included in the description. This PCS Vector is compared with those of the currently running instances. If one of them matches, then no new instantiation takes place but rather the found instance is used. If no running service instance matches the new architecture description, BootstrapRevised is invoked to turn the architecture into a running service instance. This new service instance is assigned to the user, who may now access its functionality via requests to the Instance Manager. If the user's privacy constraint information changes later on, the process may be restarted to consider the new constraints.

5. Conclusion and future work

Software product lines have been recognized as a software development paradigm that leads to improvements in terms of software cost, productivity, quality, etc. Relatively little research focuses on the potential of their use at runtime. In this article, we showed that a PLA combined with runtime variability allows for an elegant solution to address the privacy issues in web personalization relating to the fact that privacy constraints may be different for each individual user. Our approach offers the following benefits:

1. **Generality**: different types of privacy constraints are addressed in a unified way.
2. **Enforcement**: our approach does not only allow one to specify privacy requirements (such as in P3P [29]), but it also enforces their consequences on personalization methods.
3. **Runtime dynamics**: privacy requirements can be addressed dynamically during runtime, e.g. when users change their privacy preferences.
4. **Reusability of architectural instance**: if two or more users have the same set of privacy constraints, they will share a single instance of a personalization architecture.
5. **Update modularity**: new privacy constraints and personalization methods can be added in the PLA in a modular fashion, resulting in local update effects only.

Of course, we do not claim this to be a complete solution to all privacy issues. Our approach focuses on the architectural aspects of user-tailored privacy provisioning but does not control (let alone enforce) what and how user data are collected by the different user modeling components.

Ackerman pointed out a “social-technical gap” between human activities or decisions (which are inherently flexible and nuanced) and what we can support technically [37]. Such a gap also exists when we deal with people’s privacy because no system can ever know all potential user privacy constraints in advance. One future task is to conduct a user study to solicit personal privacy preferences of real users. Ensuring that our system can handle the most common privacy constraints would greatly help bridge the gap.

While we currently use a set of Boolean variables to express identified privacy constraints, ultimately these constraints should be expressed in a privacy constraint specification language such as APPEL [38] or EPAL [39], or semantic web technologies [40].

Last but not least, performance and scalability are of critical interest in practice. We need to determine empirically whether our system is able to manage architectures that provide personalization services to hundreds of thousands of users from all over the world. Since the number of privacy jurisdictions is limited (currently to about 40 countries and 100 states), we assume that many of our users will share the same architecture. The resource-intensive architecture selection and instantiation process is therefore likely not to be invoked too often. This reusability is key to performance and scalability, but its effects will need to be more thoroughly tested. We are currently evaluating the performance and scalability of our approach.

Acknowledgements. We thank Eric Dashofy, Ping Chen and Chris van der Westhuizen for their discussions on the above material and for their support of ArchStudio 3.0. We also thank Yun Huang, Suzanne Schaefer, Norman Su and our anonymous reviewers for their comments on the paper. This research has been supported through NSF grant IIS 0308277, and the preparation of the paper by a Humboldt Research Prize.

6. References

[1] C. Atkinson, J. Bayer, C. Bunse, E. Kamsties, O. Laitenberger, R. Laqua, D. Muthig, B. Paech, J. Wust, and J. Zettel, *Component-based Product Line Engineering with UML*. New York, New York: Addison-Wesley, 2002.

[2] J. Bosch, *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*. New York: Addison-Wesley, 2000.

[3] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*. New York, New York: Addison-Wesley, 2002.

[4] A. Corbett, M. McLaughlin, and K. C. Scarpinatto, "Modeling Student Knowledge: Cognitive Tutors in High School and College," *User Modeling and User-Adapted Interaction*, vol. 10, pp. 81-108, 2000.

[5] F. Linton and H.-P. Schaefer, "Recommender Systems for Learning: Building User and Expert Models through Long-Term Observation of Application Use," *User Modeling and User-Adapted Interaction*, vol. 10, pp. 181-208, 2000.

[6] L. Strachan, J. Anderson, M. Sneesby, and M. Evans, "Minimalist User Modelling in a Complex Commercial Software System," *User Modeling and User-Adapted Interaction*, vol. 10, pp. 109-146, 2000.

[7] D. Billsus and M. J. Pazzani, "User Modeling for Adaptive News Access," *User Modeling and User-Adapted Interaction*, vol. 10, pp. 147-180, 2000.

[8] A. Kobsa, "Adapting Web Information to Disabled and Elderly Users (Invited Paper)," Proceedings of WebNet-99, 1999, pp. 32-37.

[9] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an Open Architecture for Collaborative Filtering of Netnews," Proceedings of ACM Conference on Computer Supported Cooperative Work, 1994, pp. 175-186.

[10] A. Kobsa, "Tailoring Privacy to Users' Needs (Invited Keynote)," in *User Modeling 2001: 8th International Conference*, M. Bauer, P. J. Gmytrasiewicz, and J. Vassileva, Eds. Berlin - Heidelberg: Springer Verlag, 2001, pp. 303-313.

[11] A. Kobsa, J. Koenemann and W. Pohl, "Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships.," *The Knowledge Engineering Review*, vol. 16, pp. 111-155, 2001.

[12] FOR, "The Privacy Best Practice," Forrester Research, Cambridge, MA 1999.

[13] R. Hof, H. Green, and L. Himmelstein, "Now it's YOUR WEB," *Business Week*, October 5, pp. 68-75, 1998.

[14] PC, "Personalization & Privacy Survey." Edgewater Place, MA: Personalization Consortium, 2000.

[15] M. DePallo, "AARP National Survey on Consumer Preparedness and E-Commerce: A Survey of Computer Users Age 45 and Older", AARP, 2000.

[16] IBM, "IBM Multi-National Consumer Privacy Survey," IBM, 1999.

[17] M. Teltzrow and A. Kobsa, "Impacts of User Privacy Preferences on Personalized Systems: a Comparative Study," in *Designing Personalized User Experiences for eCommerce*, C.-M. Karat, J. Blom, and J. Karat, Eds. Dordrecht, Netherlands: Kluwer Academic Publishers, 2004, pp. 315-332.

[18] Y. Wang and A. Kobsa, "Impacts of Privacy Laws and Regulations on Personalized Systems," Proceedings of PEP06, CHI06 Workshop on Privacy-Enhanced Personalization, 2006.

[19] DE-TS, "German Teleservices Data Protection Act," 1997.

[20] A. v. d. Hoek, "Design-Time Product Line Architectures for Any-Time Variability," *Science of Computer Programming, special issue on Software Variability Management*, vol. 53, pp. 285-304, 2004.

[21] J. Magee and J. Kramer, "Dynamic Structure in Software Architectures," Proceedings of The 4th ACM SIGSOFT Symposium on Foundations of Software Engineering, 1996, pp. 3-14.

[22] P. Oreizy, N. Medvidovic, and R. N. Taylor, "Architecture-based Runtime Software Evolution," Proceedings of The 20th International Conference on Software Engineering, 1998, pp. 177-186.

[23] J. C. Georgas, A. v. d. Hoek, and R. N. Taylor, "Architectural Runtime Configuration Management in Support of Dependable Self-Adaptive Software," Proceedings of The 2005 Workshop on Architecting Dependable Systems, 2005, pp. 1-6.

[24] A. Garg, M. Critchlow, P. Chen, C. v. d. Westhuizen, and A. v. d. Hoek, "An Environment for Managing Evolving Product Line Architectures," Proceedings of International Conference on Software Maintenance, 2003, pp. 358-367.

[25] P. Chen, M. Critchlow, A. Garg, C. v. d. Westhuizen, and A. v. d. Hoek., "Differencing and Merging within an Evolving Product Line Architecture," Proceedings of The Fifth International Workshop on Product Family Engineering, 2003, pp. 269-281.

[26] A. Kobsa and J. Schreck, "Privacy through Pseudonymity in User-Adaptive Systems," *ACM Transactions on Internet Technology*, vol. 3, pp. 149-183, 2003.

[27] 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*, pp. 31ff, 1995.

[28] A. S. Patrick and S. Kenny, "From Privacy Legislation to Interface Design: Implementing Information Privacy in Human-Computer Interfaces," in *Privacy Enhancing Technologies*, vol. LNCS 2760, R. Dingledine, Ed. Heidelberg, Germany: Springer, 2003, pp. 107-124.

[29] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle, "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation 16 April 2002," 2002.

[30] S. Buffett, K. Jia, S. Liu, B. Spencer, and F. Wang, "Negotiating Exchanges of P3P-Labeled Information for Compensation," *Computational Intelligence*, vol. 20, pp. 663-677, 2004.

[31] S. Preibusch, "Personalized Services with Negotiable Privacy Policies," Proceedings of PEP06, CHI 2006 Workshop on Privacy-Enhanced Personalization, 2006.

[32] A. Kobsa, "Generic User Modeling Systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Heidelberg, Germany: Springer Verlag, forthcoming.

[33] A. v. d. Hoek, M. Mikic-Rakic, R. Roshandel, and N. Medvidovic, "Taming Architectural Evolution," Proceedings of The Sixth European Software Engineering Conference (ESEC) and the Ninth ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9), 2001, pp. 1-10.

[34] NAI, "Self-Regulatory Principles for Online Preference Marketing by Network Advisers," Network Advertising Initiative, 2000.

[35] ArchStudio, "ArchStudio 3.0," 2005, http://www.isr.uci.edu/projects/archstudio/.

[36] E. M. Dashofy, A. v. d. Hoek, and R. N. Taylor, "A Comprehensive Approach for the Development of XML-Based Software Architecture Description Languages," *ACM Transactions on Software Engineering and Methodology*, vol. 14, pp. 199-245, 2005.

[37] M. S. Ackerman, "The Intellectual Challenge of CSCW: The Gap between Social Requirements and Technical Feasibility," *Human-Computer Interaction*, vol. 15, pp. 179-203, 2000.

[38] L. Cranor, M. Langheinrich, and M. Marchiori, "A P3P Preference Exchange Language 1.0 (APPEL1.0): W3C Working Draft 15 April 2002," 2002.

[39] M. Schunter and C. Powers, "The Enterprise Privacy Authorization Language (EPAL 1.1): Reader's Guide to the Documentation." IBM Research Laboratory, 2003.

[40] F. L. Gandon and N. M. Sadeh, "Semantic Web Technologies to Reconcile Privacy and Context Awareness," *Journal of Web Semantics*, vol. 1, pp. 241-260, 2004.