Lightweight Software Product Line Based Privacy Protection Scheme for Pervasive Applications

Zakwan Jaroucheh, Xiaodong Liu, Sally Smith
School of Computing
Edinburgh Napier University
Edinburgh, United Kingdom
{z.jaroucheh, x.liu, s.smith}@napier.ac.uk

Huiqun Zhao
Department of Computer Science
North China University of Technology
Beijing, China
zhaohq6625@sina.com

***Abstract*— Protecting user's privacy is one of the main concerns for the deployment of pervasive computing systems in the real world. In pervasive environments, the user context information is naturally distributed among different spatial or logical domains. Many efforts have been done to match the service privacy policy with the user's privacy preferences. However, since the pervasive environments are characterized by a large number of available services as well as a large amount of context information, the privacy protection mechanism poses two main requirements. Firstly, policies are created on a per task basis. We argue here that specifying the privacy on a per domain basis facilitates specifying the privacy preferences for the user. Secondly, to ease specifying the user' privacy preferences, an intuitive mechanisms to specifying the context information that can be consumed by services are thus needed. In this paper, and in order to bridge the gap of the context information perception by the developers and by the users, we propose to represent the available context information in each domain as a feature model. In this way, the developers are able to configure this feature model to get the context information they need; the users can easily specify the context features they are willing to share. The result is a domain-oriented user-centric privacy protection scheme.**

Keywords-privacy; pervasive application; software product line.

I. INTRODUCTION

In pervasive environments the user is surrounded by a large number of devices which cooperate together to create a context-aware environment that supports her in everyday activities, e.g., business, health care, or education. In this respect, the user will enjoy a new experience in a non-obtrusive way as the existing infrastructures will be more proactive and dynamically adaptable to current situations; user preferences; and environmental context in a less intrusive way [1]. However, since the pervasive environments discover and take advantage of contextual information (such as user activity, location, time of day, nearby devices) to make decisions about how to dynamically provide services to meet user requirements, the user privacy protection and enforcement naturally becomes a main concern and obstacle prohibiting the wide spread of the pervasive environment paradigm.

Most of the privacy efforts in the field of pervasive computing have been concerned with integrating access control mechanisms into pervasive computing infrastructure (e.g. [2][3]), modeling privacy policies and their enforcements [4], employing conventional encryption and security mechanisms, or providing identity management tools (such as anonymity and pseudonymity techniques) to complete privacy protection [5]. Despite the impressive efforts in this field the existing approaches suffer from some limitations as follows.

Firstly, in this paper, a domain is defined as a logical abstraction of a spatial area or a logical concept which has a clear boundary. The context information available in each domain is managed by a separate context manager. While moving, the user roams across domains. In addition, each domain may maintain its own sensors and mechanisms for inferring context related to this user. The existing solutions address parts of privacy challenges faced in context-aware systems by letting the service providers create and enforce their privacy policy. However, as the user finds himself overwhelmed by the large number of services, specifying her privacy preferences on the basis of each task or service becomes a daunting task. Thus we propose that the user specify her privacy preferences on a per domain basis and therefore it is the responsibility of the context manager in each domain to guarantee the user's privacy. Of course, the user should be able to create different privacy preferences according to the service task or role.

Secondly, most of the existing approaches do not support the active participation and choice of individuals to control over their context information or personal data. This is due in part to the user's lack of knowledge of internal context information representations and semantics. Therefore, because the user context information needed by a service may be large, the user finds it difficult to specify her preference for access control over every piece of context information. In addition, system developers have little programming support in creating user interfaces that are effective in helping end-users manage their privacy. Consequently, the active user participation in controlling

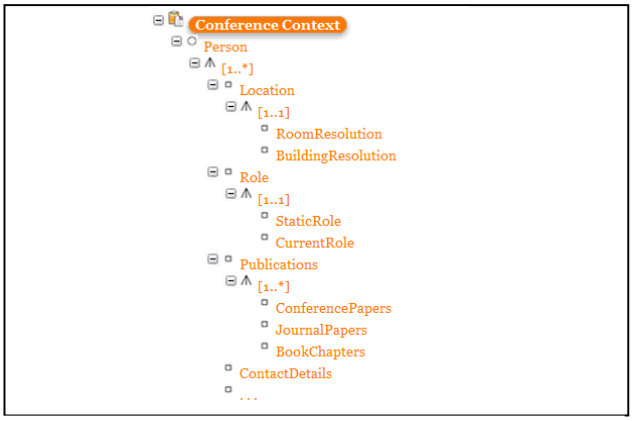

Figure 1. Context feature model

context information disclosure demands for flexible mechanisms and user interface which represent the available context information in a top-down logical fashion that allows specifying privacy requirements.

Finally, as Weiser noted, "The problem, while often couched in terms of privacy, is really one of control. If the computational system is invisible as well as extensive, it becomes hard to know what is controlling what, what is connected to what, where information is flowing, how it is being used…and what are the consequences of any given action" [6]. Under this perspective, and in order to efficiently enforce the user privacy requirements across domains, the context manager in each domain should protect user's context information over different levels of granularity.

In this paper, we propose a lightweight privacy enforcement framework which provides privacy mechanisms that allow developers and end-users to support a spectrum of trust levels and privacy needs. It is a lightweight because it does not provide a comprehensive privacy enforcement solution as this requires a combination of technology, legislation, corporate policy, and social norms [4]. It does however, provide a technical foundation for privacy-sensitive pervasive computing, making it easier for developers to build privacy-sensitive applications while minimizing the risk to people's privacy. This framework is designed such that context information is captured, stored, and processed on the context manager in each domain. Afterwards, end-users can choose what information and its granularity to share with others, thus providing greater control over their information disclosures. To achieve this aim, ideas from Software Product Line have been leveraged to represent the context information.

The rest of this paper is organized as follows. First, context information modeling is described in Section II. In Section III, we present a motivation scenario. The proposed approach is explained in Section IV and V. Related work and conclusions end the paper.

```
<configuration model="Context Feature Model">
   <feature id="Person">
     <value>1</value>
   </feature>
   <feature id="Location">
     <value>1</value>
   </feature>
   <feature id="RoomResolution">
     <value>1</value>
   </feature>
   <feature id="BuildingResolution">
     <value>0</value>
   </feature>
    ...
</configuration>
```

Figure 2. Example of context feature model configuration

II. SPL BASED CONTEXT MODELLING

In our previous work [7][8][9], we have presented an approach for context-aware software development based on a flexible product line based context model which significantly enhances reusability of context information by providing context variability constructs to satisfy different application needs. Commonality and variability management techniques from software product line have been applied to handle context variabilities for per-application customization. On the other hand, feature modeling is a key concept in product line engineering. Based on the context feature model (CFM), specific context –i.e. member of a product line– can be constructed by composing features from context information. The result is a more intuitive way to represent context and improve overall systems performance.

From the context modeler usability perspective, SPL-based context modeling allows her to think about the context information from different perspectives and use the feature model available tools. In fact, it is possible to think about the context information from different perspectives and design different feature models. In addition, CFM allows the context modeler to devise context-specific features that can be shared among all applications that need to use this context.

We propose here to extend the context feature model idea to address key concern of preserving privacy in context-aware pervasive computing environments: privacy management (i.e. allowing people to express their privacy preferences and manage privacy levels).

III. MOTIVATION SCENARIO

Alice is a researcher going to attend a conference. Once she has arrived at the conference building, she decides to contact expert researchers. Because the expertise of a researcher could be interpreted in different ways (e.g., depending on her publication in journals, on her patents or award, etc.), the conference's available context information is promoted via a CFM, which models the context variability. Conference advisor application suggests researchers that could be interesting for Alice. Fig. 1 shows a

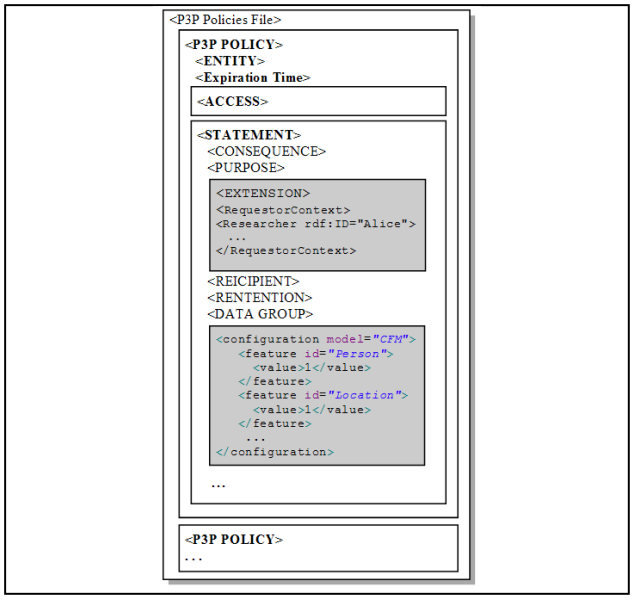

Figure 3. P3P policies file skeleton and its extensions

sample conference CFM. The CFM configuration represented in Fig. 2 is used to retrieve the locations (in room resolution) of the persons existing in the conference venue. Obviously different conference attendees have different privacy requirements specific to the conference domain.

IV. PRIVACY VOCABULARY

The context information collecting policy of the context-aware application, the privacy preferences set by the user, and the disclosure agreement between the user and the application are all expressed with a shared set of privacy vocabulary. The privacy vocabulary consists of an unambiguous representation of privacy data, as well as descriptions of disclosure conditions of the privacy data, by which both parties (the user and the application) and privacy-related functional components involved in our architecture (i.e. the context manager and the privacy matching component described later) could have a common understanding about privacy requirements while interacting with one another.

The privacy vocabulary used in our approach has been developed based on the terminology and policies specified in Platform for Privacy Preferences Project (P3P) [10], and adopted P3P policies as a basic data format in privacy data exchanges. The reason is to take advantage of the substantial legal and social expertise that has been put into the development of the P3P standards. However, since the P3P is initially an attempt to provide privacy mechanisms for the Web, it only considers a person's identifying information (such as name, birthday, credit card details, etc.) as private data to be protected. In context-aware environments, dynamically changing contextual information (such as a user's location) is also sensitive, but is not covered by the P3P specification. Thus, some extensions are necessary to P3P base data schema and regular policy elements before

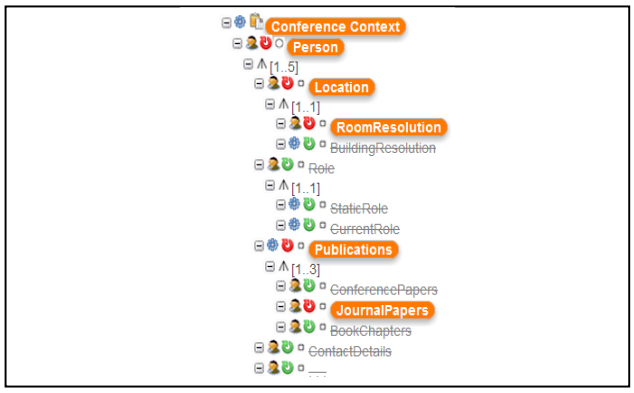

Figure 4. Example of CFM configuration

P3P could be adopted in context-aware pervasive computing environments.

In particular, similar to data schema in the P3P specification which is structured hierarchically (by using a dotted notation, such as user.home-info.telecom.telephone), we define the context features to capture the multi-level hierarchy of context information by using the CFM configuration. That is, the application can specify the context features it needs and the user has to specify his preferences to allow or disallow the application to collecting theses features in different conditions. In addition, we extend P3P's <PURPOSE> element to enable data collectors (i.e. context-aware applications) to explicitly describe their context (in other words, the type of service they offer, the name and context of the requestor, etc.). Fig. 3 shows a high-level skeleton of the P3P policies file that is used in privacy interactions in our architecture, with the block in shadow highlighting the extensions.

V. ENFORCING USER'S PRIVACY

The privacy vocabulary includes both privacy data elements and disclosure conditions. According to P3P specification [10], the conditions can be classified based on various personal concerns including recipients of data, purposes of data collection, duration that data will be kept by recipients, a user's access privilege to his personal data once stored by recipients, and ways of handling disputes. Usually the matching between the application privacy policy and the user privacy preferences is supported by P3P rule matching languages, such as APPEL [11].

In our approach, instead of relying on APPEL, we intend to provide a simpler way for the user to express their preferences. In this respect, both the developer and user should configure the context feature mode. On one hand, the developer expresses his interest in context information by configuring the CFM. On the other hand, the user can intuitively configure the CFM once the application requests the context information related to the user, or he can choose one of already saved configurations corresponding to

```
<configuration model="Context Feature Model">
   <feature id="Person">
     <value>1</value>
   </feature>
   <feature id="Location">
     <value>1</value>
   </feature>
   <feature id="RoomResolution">
     <value>1</value>
     <condition>
       <requestorLocation>ConferenceVenue</requestorLocation>
     </condition>
     <condition>
       <requestorType>ConferenceAttendee</requestorType>
     </condition>
   </feature>
    ...
</configuration>
```

Figure 5. Example of context feature model configuration

different privacy preferences schema. The user can optionally assign a condition to the inclusion of any context feature in terms of the requestor context information.

In order to guarantee the user's privacy we need to make matching between the two CFM configurations. Obviously, we can distinguish between two cases: (i) All the context features required by the application are marked as permissible by the user. In this case, the matching routine proceeds with providing the application with the required context information. (ii) Some of the context features are forbidden by the user. In this case, the user may be notified about this mismatch and it is the user responsibility to choose either to stop interacting with the application or to choose another privacy configuration scheme.

Fig. 4 shows an example of how the user configures the CFM of the example presented in Section III. Fig. 5 shows the CFM configuration where the user assigns conditions on the inclusion of context features. For example, in Fig. 5 the user mentions that she is willing to disclose her location (room-resolution) if the requestor is an attendee of the conference and he is actually located in the conference venue. The conditions tags are transformed internally into constraints on data and concepts of the ontology-based context model represented by OWL language. These constraints are represented by the Semantic Web Constraint Language (SWCL) [12].

VI. Related Work

The development of a lightweight privacy protection mechanism is an integrated part of our ongoing effort towards developing software engineering framework for context-aware adaptive systems. During the design of the privacy-respecting context-aware architecture, we had investigated some pervasive computing prototypes and systems that were specifically designed with privacy protection in mind, such as Confab [6] by Hong, PawS [4] by Langheinrich, and Privacy solutions in AURA project [13]. Our privacy solution has been building upon their experience and attempted to empower people with appropriate mechanisms to express and manage their privacy preferences with relative simplicity, which has not been a focus of the work mentioned above.

Applying P3P practices to pervasive computing environments has been proposed by [4][14]. In particular, PawS [4] presented an informative work that adapted the P3P policies to be applicable in pervasive computing environments, which serves as an important supplement and is compatible to our work. However, there is a key difference between our work and other privacy work that has attempted to use P3P. We have been employing P3P terminology and policies, both for data collectors to state collecting policies and for individuals to express privacy preferences. On the contrary, the P3P itself and most of the privacy work built upon the P3P limited the use of P3P policies only as a vehicle for data collectors to state their collecting requirements. They must employ other preference formulation languages, such as APPEL [11], to allow users to express their privacy preferences, which is not a trivial task for them.

VII. Discussion

The proposed approach has several advantages: Firstly, configuring a single CFM bridges the gap between how the developer and the user see the available context information and let the user feel more comfortable as he is actively involved in using the technology. Secondly the users have the flexibility to specify different privacy schema to be used in different scenarios or to specify different privacy schema that corresponds to different levels of privacy enforcements. Finally, from the user usability perspective, the proposed approach is intuitive; it allows her to think about the context information in top-down fashion and from different perspectives. In addition, the proposed approach benefits from the widely spread SPL available tools (e.g. [15]) to provide the user with interfacing tools to configuring her privacy preferences.

VIII. Conclusions and Future Work

This paper has presented an attempt to develop privacy protection mechanism to simplify the individuals' task to manage their privacy requirements toward dynamic context-aware environments. The proposed privacy approach taken by our work serves as a supplement to privacy protection through conventional access control and security mechanisms. The development of the privacy vocabulary presented in this work is among the first step to provision automated preference mechanisms. We are working on modeling the CFM by using the ontologies and developing a rule-based privacy policy language to be used for expressing and reasoning on context-dependent privacy preferences. In addition, to alleviate the user from choosing the suitable scheme, all predefined schema will be linked to a specific context constraint and thus the selection will be automatic in an unobtrusive way.

REFERENCES

[1] M. Weiser, “The Computer for the 21st Century,” *Mobile Computing and Communications Review*, vol. 3, 1991.

[2] F. Gandon, “Semantic web technologies to reconcile privacy and context awareness,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, Apr. 2004, pp. 241-260.

[3] G. Zhang and M. Parashar, “Context-aware Dynamic Access Control for Pervasive Applications,” *Communication Network and Distributed Systems Modeling and Simulation Conference*, 2004.

[4] J.I. Hong and J.A. Landay, “architecture for privacy-sensitive ubiquitous computing,” *2nd International Conference on Mobile Systems, Applications, and Services*, Boston, MA, USA: 2004, pp. 177-189.

[5] M. Langheinrich, “Personal Privacy in Ubiquitous Computing Tools and System Support,” 2005.

[6] R. Gold, J.S. Brown, B. Sprague, and R. Bruce, “The origins of research at PARC,” *IBM Systems Journal*, vol. 38, 1999, pp. 693-696.

[7] Z. Jaroucheh, X. Liu, and S. Smith, “Mapping Features to Context Information : Supporting Context Variability for Context-aware Pervasive Applications,” *2010 IEEE/WIC/ACM International Conference on Web Intelligence.*, Toronto, Canada: 2010.

[8] Z. Jaroucheh, X. Liu, and S. Smith, "Recognize contextual situation in pervasive environments using process mining techniques", *Journal of Ambient Intelligence and Humanized Computing*, 2(1), Springer, 2011.

[9] Z. Jaroucheh, X. Liu and S. Smith, "A MDD-based Generic Framework for Context-aware Deeply Adaptive Service-based Processes", *The 8th IEEE International Conference on Web Services (ICWS'10)*, Miami, Florida, USA, July 2010.

[10] L. Cranor, B. Dobbs, G. Hogben, J. Humphrey, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, M. Schunter, D.A. Stampley, and R. Wenning, “The Platform for Privacy Preferences 1.1 (P3P1.1) Specification,” 2004.

[11] L. Cranor, M. Langheinrich, and M. Marchiori, “A P3P Preference Exchange Language 1.0 (APPEL1.0),” 2002.

[12] H.-J. Kim, W. Kim, and M. Lee, “Semantic Web Constraint Language and its application to an intelligent shopping agent,” *Decision Support Systems*, vol. 46, Mar. 2009, pp. 882-894.

[13] U. Hengartner and P. Steenkiste, “Access Control to Information in Pervasive Computing Environments,” *9th Workshop on Hot Topics in Operating Systems*, 2003.

[14] G. Myles, a Friday, and N. Davies, “Preserving privacy in environments with location-based applications,” *IEEE Pervasive Computing*, vol. 2, Jan. 2003, pp. 56-64.

[15] M. Mendonca, M. Branco, and D. Cowan, “S.P.L.O.T. - Software Product Lines Online Tools,” *OOPSLA Companion, ACM*, 2009, pp. 761-762.