

Intra-Application Partitioning of Personal Data

Katrin Borcea, Hilko Donker, Elke Franz, Katja Liesebach,
Andreas Pfitzmann, and Hagen Wahrig

Dresden University of Technology, Dresden, Germany
{borcea|donker|ef1|liesebach|pfitza|wahrig}@inf.tu-dresden.de

Abstract. Personalization provides users a comfortable working environment. But the necessary collection of personal data can imply privacy problems. Usual approaches to minimize privacy problems aim at separating data disclosed in different applications. However, this inter-application partitioning is not sufficient in case of large applications. Here we introduce the concept of *intra-application partitioning* of personal data by means of application-internal contexts. The description of such task-related contexts enables users to assess the linkability of their actions within the application. This approach helps users to control by themselves the linkability of their personal data.

1 Introduction

Currently, most applications aim at providing users a comfortable working environment adapted to personal needs. This goal requires to collect and to evaluate personal data describing, e.g. users' preferences and goals. Any collection of personal data, however, puts privacy at risk. Since computing systems are not perfectly secure, we cannot exclude unintended access to the data.

Hence, as less data as possible should be collected and, if possible, no personal data at all should be disclosed. Privacy-enhancing Identity Management (PIM) realizes decentralized data management and transparent data processing for users [1]. The users are enabled to partition their personal data and to control disclosure of data subsets [2, 3]. Each subset of information is called a partial identity [5], which must be unlinkable by others except their owner. Therefore, uncorrelated pseudonyms are used as identifiers for the partial identities.

Usually, PIM is used to keep data in different applications separate from each other (inter-application partitioning), or to separate actions of a user within different roles. Particularly, for applications comprising different services and/or interactions with other users, the possibility to partition personal data is needed - even for repeating actions. Within this paper, we introduce possible solutions for this *intra-application partitioning* depending on *application contexts*.

In contrast to inter-application partitioning, for intra-application partitioning the application must be aware of the partitioning and must support it. Therefore, the application needs to be modified. Additionally, we have to consider interactions between users, i.e. former actions of the user under a pseudonym, and interactions with other users (possibly) under different pseudonyms.

For the realization of intra-application partitioning, we aim at utilizing mechanisms provided by the PIM architecture currently being developed within the EU project PRIME¹. As an example application that will be executed on top of this PIM architecture [1], we will use the eLearning application BluES². It is based on a client-server architecture and comprises conflicting requirements: Personalization is important for providing a reasonable working environment. Otherwise, severe privacy threats may imply acceptance problems of eLearning.

There are some approaches known in the field of eLearning which aim at enabling users to control disclosure of personal data. The authors of [9] use privacy policies to describe which data may be disclosed to whom and under which conditions. However, the partitioning of personal data is not considered. The approach introduced in [4] considers the use of PIM for using different pseudonyms for different services, but it neglects partitioning within one application.

In Sec. 2, we give a short overview of our example application. Afterwards, we introduce our definitions for contexts and general aspects in Sec. 3. In Sec. 4 we discuss possible realizations. Finally, Sec. 5 concludes and gives an outlook.

2 Short Overview on our Example Application

2.1 General Overview on BluES Concepts

eLearning comprises a number of use cases. Users can act within different roles in the eLearning environment. Within this paper, we refer to the use cases provided by BluES [1] which support *learning*, *authoring*, and *tutoring* (Fig. 1).

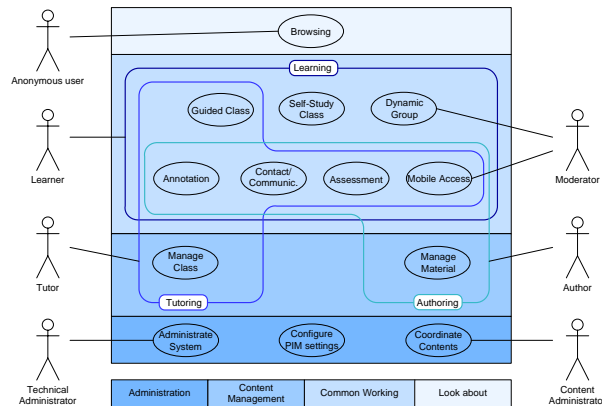


Fig. 1. Overview on use cases within the eLearning application BluES.

¹ <http://www.prime-project.eu.org>

² Java based eLearning environment, <http://www.blues-portal.de>

Our use cases can be classified into four categories: administration, content management, common working, and look about (anonymous browsing). BluES supports different learning scenarios: *guided class* (i.e. assistance by tutors), *self-study class*, and *dynamic groups* (i.e. support of cooperative learning). Common working provides additionally functionality, e.g. annotations and communication. Nearly all of the use cases shown in Fig. 1 comprise a number of sub use cases. Thus, a guided class contains, e.g. the sub use cases registration, process learning modules, practice, and examination³.

The additive use case *Configure PIM settings* enables users, e.g. to configure the partitioning of personal data and, thereby, to control the disclosure of personal data by themselves as required in Sec. 1.

2.2 Personalization and Privacy Requirements

The eLearning environment should automatically adapt the content presentation and generate hints depending on the learner's progress. This assistance implies acquiring of personal data such as learning history and his preferences.

Cooperative scenarios pose special requirements on personalization. If a dynamic group is established, users who are possibly also interested in the topics addressed in this group should get a hint from the system. Therefore, the application needs information about the interests of the users for this support.

The application as well as assisting tutors can provide the best support if they to recognize the users and if they know as much as possible about them. Even if suitable user models reduce the amount of necessary information [8], collecting and evaluating of personal data is inevitable implying that actions of a user are linkable. This linkability facilitates to create detailed user profiles [1]. Finally, those profiles allow to draw conclusions about the user (e.g. about his habits, equipment, or social status) possibly leading to a biased environment.

Therefore, a fine-grained partitioning of personal data within the application itself is essential in order to enable unrestricted behavior of users.

3 Contexts and Context Switches

3.1 Basic Considerations

Within BluES, contexts are used as means to realize intra-application partitioning of personal data. The term **context** was first introduced in [7]: Computation does not occur at a single location in a single context but rather spans a multitude of situations and locations. We use this term in a similar way to describe a specific situation in which a user works to perform a task. The application must be able both to detect the current context and to determine which actions (based on this context information) the users will take. Context information is derived

³ A more detailed description of the use cases and sub use cases can be found at <http://blues.inf.tu-dresden.de/concepts/useCases.html>.

from diverse information sources, such as user actions. A **context switch** means that the user starts working on another task.

We assume that differentiating contexts is intuitive to each user, since this simply means to differentiate tasks. If a user starts to work on a new task, he can scrutinize which other users will recognize him, what they possibly already know about him, and which information about himself he is willing to disclose to them. This implies the need for a privacy-aware user interface.

Possible context switches are predetermined by the functional structure of the application. Any action that does not belong to the current (sub) use case implies a context switch that is a potential point to change the pseudonym (switch of partial identity). Actual pseudonym switches depend on users' privacy desires as well as on application restrictions specified by means of policies.

Even if users decide to switch to another partial identity, we have to consider linkable partial identities. An observer who can see all pseudonyms at server side can at least conclude there might be pseudonyms which belong to one and the same user. Users acting at the same time in the same way establish an anonymity set [5]. The size of such anonymity sets restricts the achievable privacy: the larger the anonymity set – the smaller the degree (the probability) being identified.

3.2 A First Approach of Modeling Contexts

For an automatic evaluation of potential context switches, first, we need a description of context and context switch. A context C_i is described by a set of attributes. Two contexts must differ in at least one of these attributes. Generally, there are four classes of attributes: a general use case description uc_g (top level use case, e.g. guided class = "Advanced Statistics"), an additional use case description uc_a (sub use case(s), e.g. synchronous learning = "Hypothesis"), a history h describing actions performed within the corresponding sub use case (e.g. pose_a_question), and a description of communication partners cp :

$$\begin{aligned} C_i &= (uc_g, \{uc_a, \{h, \{cp\}^*\}^*\}) \\ uc_g &= (role, top_level_use_case) \\ uc_a &= (sub_use_case_chain) \\ h &= (date, time, action, \dots) \end{aligned}$$

A change of one of the attributes of uc_g implies a context switch in any case. If the user wants to pool different sub use cases within a top level use case, uc_a contains a list of the included sub use cases. If a user wants to separate sub use cases, uc_a contains only one sub use case. The finest possible granularity – *atomic contexts* – can be achieved if each action within a sub use case is separated. In that case, uc_g , uc_a , and h include single values only.

This context description depends on the chosen granularity which attributes are evaluated in order to recognize a context switch. These attributes are called *differentiating attributes*, which comprise at least uc_g . They also include uc_a , if the user wants to separate different use cases, and h , if the user even wants to separate actions performed within a sub use case. Since we assume that the user

does not perform different actions at the same time, *cp* is not used to distinguish contexts. However, *cp* provides linkability information to the user.

3.3 Describing Context Switches

In fact, only actions of a user that initiate communication with the server require a decision whether a context switch is implied. We call these actions *visible actions* since they are visible outside the client machine.

In case of a visible action, the application evaluates the current context of the user and the invoked action. Since any invocation includes information about the corresponding use case, the application can check the differentiating attributes. We distinguish between the following cases:

- *Mandatory context switch*: The current context and the visible action differ in at least one of the attributes of uc_g .
- *Potential context switch*: The current context and the visible actions differ in at least one of the differentiating attributes, but not in an attribute of uc_g .
- *Actual context switch*: According to the result of the evaluation, the visible action actually implies a context switch. Mandatory context switches always result in an actual context switch.

We distinguish between potential and actual context switches since we have to consider different possibilities to realize context switches. The scope of a pseudonym can comprise one or more contexts. However, the suggested structure for the description of contexts provides a reasonable basis for partitioning.

Due to the monotony of anonymity (confidentiality goals can only decrease over time [6]), it is not possible to split scopes of pseudonyms *ex post*. The linking between the newly introduced partial identities is already known to the others. Thus, context switches within a top level use case are only necessary if the user wants to change his pseudonym at this point.

3.4 Handling Context Switches

Basically, the application has to use a general event framework. Each invocation of a function, i.e. an action, is handled as event. Visible actions initiate an evaluation of potential context switches. Potential context switches can

- directly imply an actual context switch,
- imply an actual context switch as well as a notification for the user about this context switch, or
- initiate an interaction with the user where the user has to decide by himself whether an actual context switch should be initiated.

Configurations as well as already created contexts are managed and stored solely at client side. The server is not able to link different partial identities which are generated within the application. This approach enables users to control which of their data can be linked together.

4 Configuring Potential Context Switches

4.1 Possible Approaches

Basically, we can distinguish three fundamental strategies to reasonably configure potential context switches: prospective, collateral, and retrospective. Each of these concepts is based on the definition of atomic contexts.

Prospective configuration. Before working with the application, the user defines own contexts and points to switch to another partial identity. Afterwards, the defined contexts are valid for the whole session. On basis of these settings the system switches the contexts as well as the partial identities without user notification while he is working with the application.

Collateral configuration. In contrast to the first approach, every visible action requires a one-time decision: The user has to decide if he wants to switch to another context as well as to another partial identity.

Retrospective configuration. Every visible action directly implies a context switch and a switch of the partial identity. The user decides retrospectively which of these contexts and partial identities shall be linked together. This approach delivers indications for reasonable context classifications as well as for supporting especially cooperative learning.

Table 1 discusses advantages and disadvantages of these approaches.

Configuration Approach	Advantages	Disadvantages
Perspective configuration	Users are not disturbed in their working and thinking processes by system queries concerning possible context switches.	Not suitable for unexpected and new situations; users should be very familiar with the application as well as with privacy aspects.
Collateral configuration	Offers the best flexibility; is particularly suitable for personal adaptations to current requirements.	Users are disturbed each time there is a possible context switch.
Retrospective configuration	Per default, this approach provides the best anonymity at run time.	The user must actively analyze his data track backwards in order to summarize possible contexts.

Table 1. Advantages and disadvantages of the different configuration approaches.

To conclude, none of the strategies can be used stand-alone. Only a combination of the three concepts is reasonable. Particularly, an eLearning environment is too complex for perspective configuration. Unknown situations must always imply interactions with the user. Furthermore, there should be the possibility to analyze and to link contexts as well as partial identities backwards in order to derive strategies for future decisions.

Fig. 2 gives an overview of the three different concepts. The example scenario illustrates that the user has access to his personal workspace in the scope

of pseudonym "Ps 1". Further, he attends the self-study class "Principles of Statistics" and the guided class "Advanced Statistics". In the latter, he separates different actions using the pseudonyms "Ps 2" and "Ps 3".

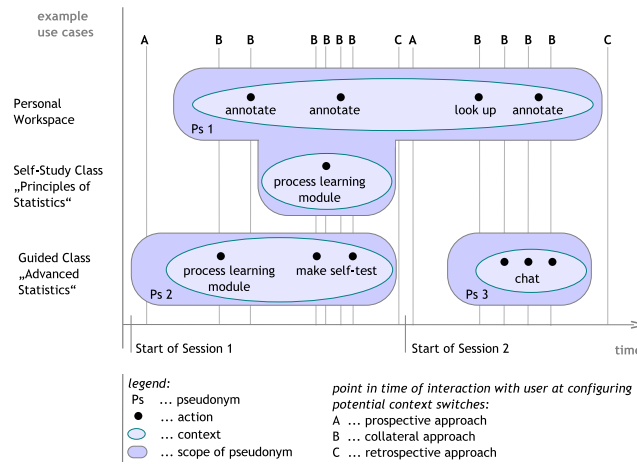


Fig. 2. Three approaches of configuring potential context switches.

4.2 Supporting Users in Defining Context Switches

Even if we assume that differentiating tasks is intuitive for users, configuring potential context switches might be not easy for them, especially, if users are not yet familiar with the application or with privacy issues. Therefore, support for users is an essential criteria for the acceptance of intra-application partitioning.

Thus, reasonable pre-configurations for context switches should be provided (context switch with/without notification, or interaction). Users could decide to download such privacy settings from a trusted provider and integrate into their application. When the users become more familiar with the application structure and with data partitioning they are enabled to adapt the default settings to their actual needs. The configurations should be possible for types of use cases as well as for specific use case instances. For example, users can define rules for guided classes but overwrite these general rules for a specific guided class they attend.

Of course, the user interface of the application must be enhanced in order to display the user's current privacy status.

5 Summary and Outlook

We described possibilities for intra-application partitioning of user data. This approach enables users to preserve their privacy even in applications which support

interaction with other users. The application itself must enable this partitioning. Therefore, we characterized application contexts used for the partitioning. The application should support flexible configuration of potential context switches. In order to reasonably support users we suggested to deliver pre-configurations.

There are still many open issues: Assessing linkability requires thoroughly further investigations. Within this paper, we have only discussed contexts and context switches from the point of view of a single user. However, multilateral interactions in cooperative scenarios will pose further requirements that concern a set of users. Furthermore, we have to consider linkability due to observation of context switches.

Another topic of future work is visualization of linkability, i.e. developing a suitable user interface, that helps users to decide whether they should generate a new partial identity or select an existing one. This includes information about other users as well as information about the privacy state of the user himself.

Currently, we are realizing the approach described in this paper. Our goal is to perform test trials with a reasonable user set in order to investigate performance aspects as well as user acceptance and necessary effort.

We like to thank Mike Bergmann, Sebastian Clauß, and Thomas Kriegelstein for helpful discussions. The work reported in this paper was supported by the IST PRIME⁴ project; however, it represents not necessarily the view of the project.

References

1. K. Borcea, H. Donker, E. Franz, A. Pfitzmann, and H. Wahrig. Towards Privacy-Aware eLearning. Accepted for PET 2005.
2. S. Clauß and M. Köhntopp. Identity Management and its Support of Multilateral Security. *Computer Networks*, (37):205-219, 2001.
3. S. Clauß, A. Pfitzmann, M. Hansen, and E. V. Herreweghen. Privacy-Enhancing Identity Management. The IPTS Report 67, pages 8-16, Sept. 2002. <http://www.jrc.es/pages/iptsreport/vol67/english/IPT2E676.html>.
4. T. Klobučar, V. Seničar, and B. J. Blažič. Privacy and personalization in a smart space for learning. *Int. J. Cont. Engineering Education and Lifelong Learning*, 14(4/5):388-401, 2004.
5. A. Pfitzmann, M. Hansen: Anonymity, Unobservability, Pseudonymity, and Identity Management — A Proposal for Terminology. Draft status: http://dud.inf.tu-dresden.de/literatur/Anon-Terminology_v0.21.pdf; September 2004.
6. A. Pfitzmann and G. Wolf. Properties of Protection Goals and their Integration into a User Interface. *Computer Networks*, 32:685-699, 2000.
7. B. N. Schilit, N. Adams, and R. Want. Context-Aware Computing Applications. In *Proc. of the 1st Int. Workshop on Mobile Computing Systems and Applications*, pages 85-90. IEEE, 1999.
8. J. Self. Bypassing the intractable problem of student modelling, 1988.
9. G. Yee and L. Korba. Privacy Policies and their Negotiation in Distance Education. In P. Derbyshire, editor, *Instructional Technologies: Cognitive Aspects of Online Programs*. IRM press, 2004.

⁴ The PRIME project receives research funding from the European Community's 6th Framework Programme and the Swiss Federal Office for Education and Science.