Delivery of Consistent and Integrated User's Data within a Multi-Tenant Adaptive SaaS Application

Aivars Niedritis

University of Latvia, Faculty of Computer Science, Raina boulv.19, Riga
Aivars.Niedritis@lu.lv

Abstract. The adaptation architecture for Web information systems (WIS) is described, which ensures two levels of adaptation – coarse-grained adaptation for the organization level and fine-grained adaptation for the user level. The situation, when users can work with many instances of the system adapted for different organizations, is also supported. This paper describes in detail the user level adaptation and particularly the adaptation of the content according to user's profile. Problems concerning data storage, integration and delivery to the user are discussed and solutions within this adaptation architecture are proposed.

1 Introduction

The requirements for Web applications stated in [1] include among others also customization possibility and dynamic adaptation of content structure.

In [2] it is defined that a Web-based Information System (WIS) is adaptive if it can personalize delivery of contents and services according to the context of the client. Personalization requires the adaptation of applications according to the preferences of the user and the context of the user [3].

Many architectures and approaches [2], [3], [4] exist for the WIS adaptation. One of the approaches is PAM (personalized access model) [3]. PAM allows performing different personalization scenarios, e.g. preference-based recommendation. In [2] an approach to the adaptation of the Web information system according to the context is presented. Authors of [4] present architecture for the personalized presentation layer of WIS to support navigation and different views on the presented data.

If we want to deliver to the user the most appropriate WIS, we should also consider approaches to the development of flexible and cost effective systems, e.g. Software product line approach (SPL) [5] or Software as a service (SaaS) approach [6].

SaaS is defined [6] as a "software deployed as a hosted service and accessed over the Internet". SaaS services can be provided to organizations of different size to support business processes common for these organizations. The organizations subscribe to use the software and pay for the usage. The SaaS application is hosted on vendor's servers.

SaaS applications can be provided at four different maturity levels [6], and one of them is multi-tenant single instance solution. The multi-tenancy means that users

from different organizations use the same application instance, but data are distinguished between tenants of the service.

In case of SaaS approach the user could be also a customer of more than one organization that uses the same SaaS approach based WIS. Architecture in case of multi-tenancy meets with problems how to create, update and deliver consistent data to the organizations and users.

Our architecture already proposed in [7] is based on SaaS ideas, it includes additional adaptation components. The two levels of adaptation are provided. Not only the organization, but also the users get their own adapted instance of the WIS. This paper describes in detail the detailed level adaptation and particularly the adaptation of the content. Problems concerning data storage, integration and delivery to the user are discussed and solutions within this architecture are proposed.

The rest of the paper is organized as follows. Section 2 presents the adaptation architecture of a Web Information System and describes two levels of adaptation. Section 3 describes how the integrated user interface that contains all user's data is constructed. Section 4 ends the paper with conclusions.

2 WIS Adaptation Architecture

In our previous work [7] we proposed adaptation architecture of a Web Information System (WIS). This architecture could be used for organizations in the same business domain and with the same target user group. This architecture uses the ideas of SaaS, e.g. multi-tenancy. The architecture allows adapting the initial configuration of the WIS for each particular organization and moreover – for each particular user, and supports the integration of all functions and all user data from all WIS instances, which are used by the particular user.

Let as consider a sample case when the user is a user of WIS of two different organizations.

2.1 Architecture Components

The adaptation architecture for our sample case is shown on the Figure 1. This is a simplified schema of the architecture (not all detailed components are included). The whole description of the architecture is given in [7].

The adaptation architecture of a WIS has the following components:

1. Context monitor identifies the context properties,
2. Adaptation component identifies the initial configuration of WIS and performs its adaptation in two levels:

Coarse level adaptation component finds out the function groups and functions that are accessible to the user from the configuration data according to the user's organization and the usage context, e.g. entry point (defined by the link and login data). Organization level instances of WIS are created. The organization level instances of WIS for our sample case are depicted on the Figure 1 as WIS_1 and WIS_2.

Detailed level adaptation is based on the information stored in the user-describing profiles. The detailed adaptation consists of two steps: (i) navigation adaptation and (ii) detailed adaptation of content that integrates all user data from all WIS instances.

3. WIS data layer contains user-describing profiles that are used in both adaptation levels and the business data. Data is physically stored in one database instance, but it consists from virtual data stores owned by particular organizations.

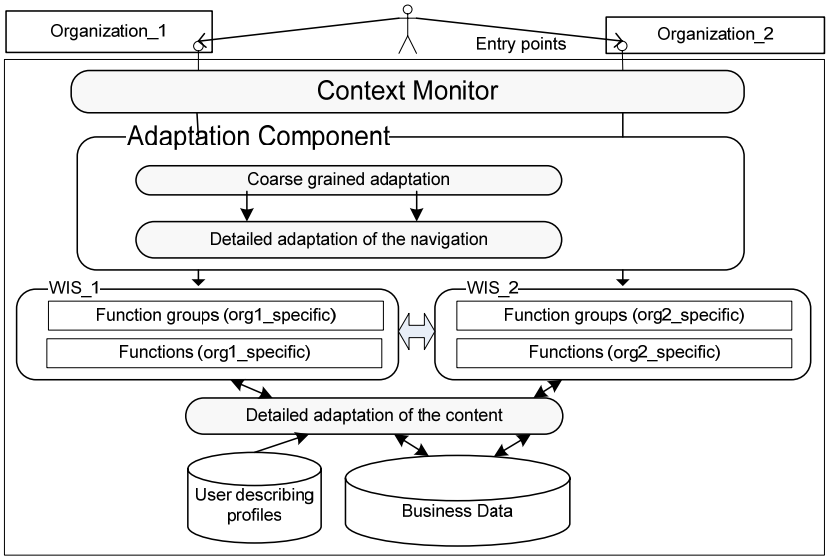

Fig. 1. Adaptation architecture of WIS

The profiles used in the proposed architecture are: configuration profile, organization profile, and user profile.

WIS configuration profile has classes *Function group*, *Function*, and *Transition*. The organization level profile describes the individual properties (e.g. local configuration, layout, etc.) essential for the organization level instance of the WIS. The elements of the model are *Organization*, *Function access*, *Accessibility*, and *Layout*. The user profile describes the affiliation of the user in the particular time period in one or more organizations using WIS and his access rights to functions and data. The elements of the user model are *User*, *Affiliation*, *Table*, and *Row_Access*. The profile uses also the class *Function* from the configuration profile and the class *Organization* from the organization profile. Model diagrams of all three profiles are given in [7].

2.2 Two Levels of Adaptation

The problems discussed later in this paper are mainly connected with detailed adaptation of the WIS, but to give an insight into the whole process of adaptation levels and corresponding operations are given. The detailed description is given in [7].

During the coarse level adaptation organization level instances are dynamically built. The adaptation operations of coarse level adaptation are following: the construction of initial configuration of WIS, the WIS adaptation in organization level, joining of instances of function groups, selection of allowed function groups, and adaptation of layout.

Afterwards the detailed level adaptation is performed based on the user profile. The detailed adaptation has two operations [7]: detailed adaptation of the navigation and detailed adaptation of the content.

The adaptation of the content corresponding to the particular user finds out his restrictions to the data, which are defined in user profile, for example, by using tables, their columns and filters for these columns. The WIS instance with the adapted navigation is supplemented with the content, which in default is defined by the data usage of the functions. During the adaptation of the content it is adjusted to the restrictions defined for the user.

As a result the user is returned an interface that integrates user's functions and user's data from all WIS instances of organizations where the user belongs to. In the next section the construction of the integrated user interface of the adapted WIS instance during the adaptation process is described in detail.

3 Construction of an Integrated User Interface

Let as consider a sample case when the user is a user of WIS of two different organizations. The interface of the adapted instance of WIS according to the results of adaptation in both levels (coarse level and detailed level) is shown on the Figure 2.

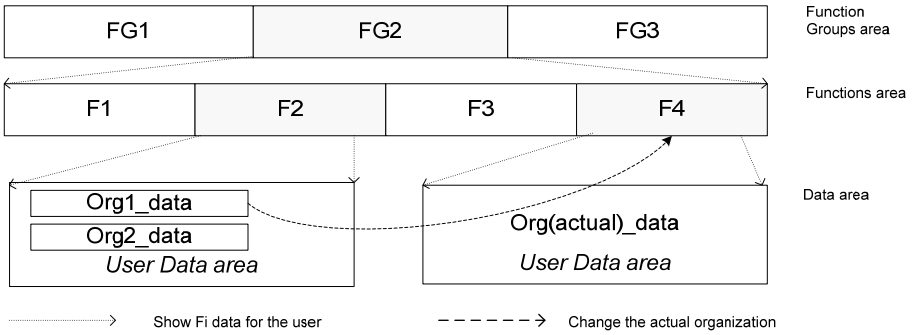

Fig. 2. WIS function groups, functions and integrated user data.

3.1 Components of the WIS User Interface

The WIS_user interface consists of following components:

1. Area of function groups (result of the organization level adaptation), in our sample case there are 3 function groups (see Figure 2). The function groups can be of two

types: (i) time-dependent function groups; (ii) static (time-independent) function groups. Both types of function groups are build during the WIS adaptation process:

- In the organization level adaptation of WIS it is detected that the particular user belongs to both organizations. From the WIS configuration profile it is determined, which function groups are defined for each organization. According to this, there can be different situations, when the particular function group is defined for both or one of the organizations. So, integrated instance *FGi_org1_org2* or instances of the function groups *FGi_org1* or *FGi_org2* are created.
- During the adaptation the time-dependent function group is dynamically built. In the configuration profile some functions could have actual_date_start and actual_date_end that determine when the function is topical for mostly all users of WIS (may be determined by the business conditions of the organizations). In this case the function is temporally excluded dynamically from the function group associated with it in the configuration profile.
- The actual function group is always shown first (function group *FG1*, Figure 2) for the user. The functions area is composed from all actual functions from both organizations of the user.

2. Area of functions (result of the organization level adaptation and also detailed adaptation of WIS, e.g. navigation adaptation) in our sample case (see Figure 2) contains 4 functions; they are shown as a result of selection of function group *FG2* by the user.

- From the WIS configuration profile it is determined, which functions are defined for selected instance of function groups (e.g. *FGi_org1_org2*, *FGi_org1* or *FGi_org2*).
- Functions can be of two types – determinant functions and dependent functions. The determinant functions are functions that work with data that determines the data used by dependent functions. For example, a determinant function "Show study data of the user" has parameters: study program, study status, study start date, etc. But the dependent function in this case could be a function "Show the grades of the user", which provides the information about grades of the user only for the study program chosen by the user in the determinant function.

3. The data area of the user (the result of detailed adaptation of WIS) in our sample case (see Figure 2) has two possible situations:

- In the case of determinant function (see Figure 2, function F2) the user can chose the data used as parameters in dependent functions. In Figure 2 it is depicted as *Org1_data* and *Org2_data*. These data elements are provided to the user in the order how they are created in WIS. We can speak about an actual data element, which usually is the last created data element. In some cases the definition of the actual data element can be more complicated, e.g. see nest section the definition of determining organization. This data is always data connected to the user, for example, study programs of the user in two universities. Let *Org1_data* is the actual data element in our sample case.
- In the case of dependent function (see Figure 2, function F4) the content of the user data area consist of data that depend on the current data element (*Org1_data* in our sample case) and is retrieved by the dependent function.

3.2 The Integration of User's Data in the User's Data Area

In previous section described sample case (see Figure 2), and also in all such applications of WIS adaptation architectures one of function groups is e.g. "Identity" and functions could be "Show image", "Show personal data".

We can speak about two types of data used by the functions that should be integrated and shown in corresponding data area. One type of data is the user-centric data that characterize the user as a person (e.g. identification information, address, education). These data could be entered by authorized users of one or other organization or by the user himself/herselfself, but this data is not connected with organizations' business functions. The other type of data is created by user or about the user during execution of organizations' business processes. For example, the student enrols to the study courses or the lecturer enters the grades obtained by the student.

The integration problems arise in the case of the first type of data. The reason of the problems is the ownership of data, when the user is a customer of the organization (e.g. student of more than one university in the long time period or may be also simultaneously).

To explain the situation let we see a motivating example. The student studied in the University1. The student's personal data was entered into the system by an authorized user. It seems that the University1 owns the student personal data (the possibility to enter and update data by user itself will be described later). The administrative staff has also documents that serve as a basis for entering users' data e.g. copies of diplomas, application forms signed by user etc. After a period of time the student applies to the University2. If the documents provided to the administrative staff of University2 (e.g. application form) contain the information conflicting with data in the system, the problem arises how to manage the conflicting information. The problem is that one data element has different values for different organizations, concerning the same person, the same attributes, but gathered from different information sources, e.g. different documents provided by the same person.

The implementation of the WIS for the universities showed also that the option that the user manages the personal information himself/herself do not satisfy the administrative staff, because some users do not pay attention to the correctness and topicality of data, the others can create data that do not correspond to documents sent neither to one university, nor the other one.

Solutions that are used to store, update and provide consistent and integrated information to the user and to the organization are following:

- Business logic dependent statuses are used to determine the data ownership in the case of data that are common to all organizations.

This solution to store and change user data is used for limited set of attributes that is chosen before to start the WIS usage and is meant for the identification of the user, who appears in the WIS regardless his organization in the current time period. The set of attributes should be minimal but enough for identification and attributes that change their values rarely are recommended.

The Table 1 shows 2 examples of records with 4 attributes that describe the user; attribute *Owner* stores the information about the organization, which created or currently owns the record. Attribute *Status* is calculated using business rules, in the

simplest case it can be checked by a trigger that detects the values of some other attributes.

For example, if the user currently studies at the university U1 (the first record, Table 1), the *status* is 'active'. In this case, if the student starts to work at other university U2, and the administrative staff at U2 wants to change the identification data of the user, they should to contact the owners of data (by phone, by WIS function that implements messages etc.).

If the user has studied at the university U2 (the second record, Table 1) and has finished his/her studies, the corresponding status of the record is 'ended'. In this case, if the administrative staff wants change these attributes, they can do it without contacting the former owner and the attribute owner is set to 'U2'.

Table 1. Data ownership examples for user identification

Name	Surname	Pers_code	Birth_date	Owner	Status	Contact
Linda	Liepina	12345	16.08.1990	U1	active	Tel.: 292929
Andris	Berzins	67899	01.01.1989	U2	ended	Tel.: 353535

- Directly stored different records for the same attribute set for the each organization, if needed.

The attributes that can have different values for each organization are stored as organization-dependent records; the attribute Owner shows the organization that has created, can update and use the record. The Table 2 shows three records about education of one user, e.g. 'Linda Liepina' (See Table 1). The records 2 and 3 show the information about the same master level education, but for both organizations (owners of the records) the values of the attribute Doc_Date differ. The first record about the bachelor level education is needed only for the organization U2, so the second record about the bachelor level education is not created for organization U1.

Table 2. Data ownership examples for organization dependent records

Nr	Level	Institution	Document No	Doc_Date	Owner
1	Bachelor	U1	11111	20.06.2008	U2
2	Master	U2	22222	10.06.2010	U2
3	Master	U2	22222	2010	U1

- One of the organizations in the WIS architecture is always defined as a determinant one. This depends on the entry point to the system (defined by link and login information).

The determinant organization is used to change the actual data element in the user data area of determining function. By default the actual data element is set to the newest one. Sometimes the identification data of the user can indicate that the organization does not have more current obligations with the user (status='ended', see Table1). Despite the data element of determinant function is the newest one, the actual data element in this case is set to the newest data element owned by the determinant organization.

4 Conclusions

This approach to the WIS adaptation use two levels of adaptation – coarse-grained for the organization and fine-grained for the user. This architecture also supports the situation, when users can work with many instances of the system. The adapted WIS organization level instances are integrated into one instance for a particular user.

The multi-tenant and single physical instance ideas used in the architecture caused the problems concerning the data ownership. This paper provides some examples from real life implementations of WIS based on adaptation architectures that illustrate the discussed problems. The proposed solutions for the data storage, update and consistent and integrated delivery to the user within the adaptation architecture also are used in the implementations of the WIS.

The described architecture is implemented and is being used in two different WIS. One of them is used in the project, where 12 universities are using each an adapted instance of the system. The other implementation is used in the Road Safety Directorate and insurance companies for the car registration WIS. So, this architecture can be used in different business areas, but it is reasonable that the organizations would share the same set of users.

Acknowledgments. This work has been supported by ESF project No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044.

References

1. Fraternali, P.: Tools and Approaches for Developing Data Intensive Web Applications: a Survey. J ACM Comput. Surv. 31(3), 227--263 (1999)
2. De Virgilio, R., Torlone, R.: A General Methodology for Context-Aware Data Access. In: Proceedings of the 4th ACM international Workshop on Data Engineering for Wireless and Mobile Access, MobiDE '05, pp. 9--15. ACM, New York (2005)
3. Abbar, S., Bouzeghoub, M., Kostadinov, D., Lopes, S., Aghasaryan, A., Betge-Brezetz, S.: A Personalized Access Model: Concepts and Services for Content Delivery Platforms. In: Kotsis, G., Taniar, D., Pardede, E., Khalil, I. (eds.) Proc. of the 10th Int. Conf. on Information Integration and Web-Based Applications and Services, iiWAS '08, pp. 41--47. ACM, New York (2008)
4. Tvarožek, M., Barla, M., Bieliková, M.: Personalized Presentation in Web-Based Information Systems. In: Leeuwen, J., Italiano, G.F., Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) Proc. of the 33rd Conference on Current Trends in Theory and Practice of Computer Science. LNCS, vol. 4362, pp. 796--807. Springer-Verlag (2007)
5. Clements, P. , Northrop, L. Software Product Lines: Practices and Patterns. Addison-Wesley, (2002).
6. Microsoft, Architecture Strategies for Catching the Long Tail, (2006)
7. Niedritis, A., Niedrite, L.: The Adaptation of a Web Information System: a Perspective of Organizations. In: Pokorny, J.; Repa, V.; Richta, K.; Wojtkowski, W.; Linger, H.; Barry, C.; Lang, M. (Eds). Information Systems Development, Business Systems and Services: Modeling and Development, Springer (2011)