A3-D10

Personal Information Systems in the Semantic Web - Lesson Learnt and Best Practices

Project title:	Reasoning on the Web with Rules and Semantics
Project acronym:	REWERSE
Project number:	IST-2004-506779
Project instrument:	EU FP6 Network of Excellence (NoE)
Project thematic priority:	Priority 2: Information Society Technologies (IST)
Document type:	D (deliverable)
Nature of document:	R (report)
Dissemination level:	PU (public)
Document number:	IST506779/Hannover/A3-D10/D/PU/a1
Responsible editors:	N. Henze
Reviewers:	I. Brunkhorst, B. Mehta
Contributing participants:	Edinburgh, Hannover, Tekniker, Turin, Warsaw
Contributing workpackages:	A3
Contractual date of deliverable:	August 31, 2007
Actual submission date:	October 1st, 2007

Abstract

This deliverable summarizes the results and experiences of working group A3 on establishing personalization strategies in the Semantic Web by presenting best practices for personal information systems in the Semantic Web. It builds on the work reported in Deliverables A3-D1 to A3-D9 which have investigated on necessary theoretical foundations (Deliverables A3-D1, A3-D4 and A3-D7), Prototypes and Practical systems (A3-D2, A3-D6, A3-D8), and architectural investigations and developments (A3-D3, A3-D5, A3-Do).

Keyword List

semantic web, reasoning, personalization

Project co-funded by the European Commission and the Swiss Federal Office for Education and Science within the Sixth Framework Programme.

Personal Information Systems in the Semantic Web - Lesson Learnt and Best Practices

F. Abel[1], A. Arnaiz[2], M. Baldoni[3], C. Baroglio[3], N. Henze[1], M. Klopotek[4], A. Martelli[3], V. Patti[3]

[1] L3S Research Center, University of Hannover, Germany
Email: abel, henze@l3s.de

[2] Fundacion Tekniker, Spain
Email: aarnaiz@tekniker.es

[3] Dipartimento di Informatica, Università degli Studi di Torino, Italy
Email: {baldoni,baroglio}@di.unito.it

[4] Institute of Computer Science – Polish Academy of Sciences, Poland
klopotek@ipipan.waw.pl

October 1st, 2007

Abstract
This deliverable summarizes the results and experiences of working group A3 on establishing personalization strategies in the Semantic Web by presenting best practices for personal information systems in the Semantic Web. It builds on the work reported in Deliverables A3-D1 to A3-D9 which have investigated on necessary theoretical foundations (Deliverables A3-D1, A3-D4 and A3-D7), Prototypes and Practical systems (A3-D2, A3-D6, A3-D8), and architectural investigations and developments (A3-D3, A3-D5, A3-Do).

Keyword List
semantic web, reasoning, personalization

Contents

1 Introduction

This deliverable summarizes the experiences and insights of the working group A3 on establishing personalization and personalized information systems on the Semantic Web. It discusses the fundamental relation between personalization and improved user experiences with digital information, and highlights the challenges and advantages that the Semantic Web offers for personalization (see Section 2). Carefully selected examples of successful and influential applications that have realized within REWERSE are summarized as *best practices – applications* (see Section 3). Crucial for personalization in dynamic and open environments like the (Semantic) Web are *effective* and *efficient* architectures: we characterize *best practices – architectures* (see Section 4). Finally, recommendations for realizing personalization are given, with respect to *functionality* (Section 5.1), *strategy* (Section 5.2), and *architecture* (Section 5.3).

2 Personalization Revisited

Personalization can be defined as the *process of optimizing an individual user's interaction with software*. The *optimization* can take place in various ways; it is not limited to certain technical or conceptual approaches, and has normally to be done at runtime. This means, a piece of software that is able to personalize to the needs of its individual users has to make informed decisions how to react in *certain situations*, in varying *conditions*, to the *needs* of the user who is currently using it. To do so, the software needs *knowledge* on the *situation*, the *conditions*, and the *needs* of the user. Whereas a characterization of the *situation*, as far as the software is concerned, can be described beforehand, the characterization of the *current situation of the user* is incomparable more difficult to obtain (e.g. one example of the difficulties that need to be mastered the work on resource-limited adaptation [Müller et al., 2001]). The *conditions* which describe the interaction scenario are not only determined by the system, but also from factors outside, and the *needs* of a user have to be guessed / derived / concluded correctly. With wrong assumptions about the current needs of the user, any personalization is likely to fail: it personalizes to non-matching needs, and even if the personalization to these wrong determined needs is good, it will only by chance satisfy the user.

The personalization task is, as indicated above, not trivial, and it relies on techniques from artificial intelligence to model and characterize the different stakeholder of the personalization process, to reason about derived and determined characteristics, and to decide on appropriate actions to take.

Personalization strongly benefits from semantic technologies, and in particular from the Semantic Web, in many ways, which will briefly be mentioned in the following points:

Advantage 1 – Machine processable semantics: The Semantic Web, with the goal to add machine processable semantics to Web resources, improves the chances for successful personalization, as the personalization process can use *reliable* information on Web resources, and *reliable* information on the relations of Web resources, to others, their intended meaning, use, and purpose.
The cutting edge, however, is, that, with the Semantic Web, *Web resources are separated from presentation / delivery*. Thus, different usage scenarios, different delivery controls are possible and supported, the re-use of content, the combination and mixing of Web resources into whichever new interaction scenarios is intended. The optimization of this plethora of usage scenarios with

Web resources according to the particular needs of a user is obvious - and is, in other words, *personalization*.

Advantage 2 – Reasoning: With the Semantic Web and the machine processable semantics of Web resources, it is possible to exploit, syndicate, and reason on Web resources. This supports the various steps to be taken during personalization, as formal descriptions and models of the stakeholder of the personalization process are facilitated. Besides the obvious technical advantages, there are other advantages for personalization: Reasoning gives the advantage of *explaining* why a certain result has been derived by a personalized system. Thus, the awareness of users of the personalization process can be improved which likely will result in better user experiences. Also, the possibilities for controllability of the personalization process increase. Lack of awareness and controllability have been in the past identified as one of the major drawbacks of current personalized systems and its improvement is a necessary steps towards successful and widely accepted systems.

Advantage 3 – Semantic Web Technologies: In the broader sense, technologies – that have been developed independently from the Semantic Web but are very handy and receive further improvement with added semantics – support personalization. In particular, service technologies will play, as we claim, an important role in future personalization. The idea is to establish *services* in the (Semantic) Web; where *services* is meant in both the literal and the technical sense:

- *Services offering personalization:* Services which carry out some task for a certain user / user group, like recommendation services, context- or location-aware services, notification services, syndication services, information broker services, scheduling services, etc. All these services have in common that they know a certain bit about their users in order to optimize the task they carry out to fit best to a certain user's needs. We call these services *personalization services*.
- *The personalization of services:* The personalized access to, and the customization of (Semantic Web) services, where the services itself might or might not be personalization services.

3 Best practices: Examples of successful *applications* offering personalization in the Semantic Web

In this section, we give an overview of approaches and applications – developed during the REWERSE project – that realize personalization functionalities based on Semantic Web technologies and reasoning.

3.1 Personal Curriculum modelling, planning and validation

Curriculum planning and validation offer useful support in many practical contexts and can fruitfully be combined with each other for helping students as well as educational institutions. Often a student knows what competency he/she would like to acquire but has no knowledge of which courses will help acquiring it. Moreover, taking courses at different Universities is becoming more and more common in Europe. As a consequence, building a curriculum might become a complicated task for students, who must deal with an enormous set of courses across the European countries, each described in different languages and on the basis of different keywords.

The need of personalizing the sequencing of learning resource, w.r.t. the student's interests and context, has often to be *combined* with the ability to check that the resulting curriculum *complies* to some abstract *specification*, which encodes the *curricula-design goals*, expressed by the teachers or by the institution offering the courses.

The *Personal Curriculum Planner* [Baldoni et al., 2006, Baldoni and Marengo, 2007] is a service-oriented personalization system, set in an educational framework, based on a semantic annotation of courses, given at a knowledge level. The system supports reasoning-based curriculum sequencing and validation:

Curriculum Planning: building personalized curricula, formalized by means of an action theory. Classical planning techniques are adopted, which take into account both the student's initial knowledge (context) and her learning goal.

Curriculum Validation: verifying the compliance of curricula w.r.t. the course design goals. Course design goals are specified in a curricula model, where the design goals formalized as a set of LTL temporal formulas expressing constraints at the knowledge level.

What A course teaches, and what is requested to be known for attending it in a profitable way, is described by means of preconditions (prerequisites) and effects (learning objectives).

3.2 Personal Publication Reader

The *Personal Publication Reader* [Baumgartner et al., 2005] (PPR) provides a personalized access to publications and allows navigating through publications within an embedded context. The PPR gathers information about publications from distributed, heterogenous sources, extracts machine-readable semantics and enriches them with knowledge about authors, etc. so that rule-based reasoning can be used to provied context-adapted access.

The publication data is extracted from several websites distributed accross Europe with the aid of the *Lixto suite*[1] and converted to RDF using *Dublin Core vocabulary*[2]. By mapping

[1] http://www.lixto.com/
[2] http://dublincore.org/documents/dces/

the authors of publications to instances of an ontology which models researchers and their involvements in organizations the PPR is able to benefit from the content and structure of the so-called *Researcher Ontology*[3]: On the one hand additional information about the authors – like mail, employer, etc. – can be presented to the user and on the other hand the structure can be used to point to related publications, e.g. publications of researchers that are involved in the same project like the author of the actual publication viewed by the user.

3.3 BEATCA

The map-based BEATCA search engine [Kłopotek et al., 2007] allows for running a full search engine cycle (spidering, indexing and analysis) on a user-selected document collection of up to 1 million documents (either local or from the Internet), creating a SOM-like document collection map and allowing for querying the collection. The user can express the preferences when collecting the documents (spider intelligence), when analyzing the collection (via parameter choice to obtain various maps from various perspectives. The system chooses then the map that is most appropriate for presentation of query results. The most important aspect for personalization is the very short time of processing (spidering, indexing and analysis) in the domain of map-based search engines which makes personal usage feasible. The next one is the capacity of incremental processing with new documents (without radical change of the view of the document collection) preventing from radical map changes, typical for other SOM approaches.

3.4 MyEar & MyNews

The *MyEar music recommender* [Abel et al., 2006] enables users to listen to personalized podcasting feeds. MyEar is realized with the aid of the Personal Reader Framework and provides a Personalization Service which searches the web for podcasting feeds that go with the users taste in music and combines items of this feed to produce a personalized podcasting feed. More precisely, the MyEar Personalization Service utilizes the Google search engine to detect RSS 2.0 feeds, which are potentially adequate, and then performs postprocessing by analyzing metadata stored within those feeds to identify items that should be included into the resulting podcasting feed. The MyEar Syndication Service, which implements the application logic of MyEar, accesses the User Modeling Service of the Personal Reader framework (cf. section 4.1) in order to store songs the user likes or to determine playlists of the user's friends etc.

MyNews is very similar to the MyEar application. It reuses parts of the MyEar application and generates personalized news feeds. More details are presented in Deliverable A3-D9.

3.5 Personalized Preference Search

The *Personalized Preference Search* application [Abel et al., 2007b] mainly focusses on retrieval of e-learning resources. While the growing number of learning resources increases the choice for learners, it also makes it more and more difficult to find suitable courses. Thus, improved search capabilities on learning resource repositories are required. With the Personalized Preference Search, we propose an approach for learning resource search based on preference queries. A preference query does not only allow for hard constraints (like 'return lectures about Mathematics') but also for soft constraints (such as 'I prefer a course on Monday, but Tuesday is also

[3] http://www.personal-reader.de/rdf/ResearcherOntology.owl

fine'). Such queries always return the set of optimal items with respect to the given preferences. The core of this application is the Personal Preference Search Service (PPSS) which offers significantly enhanced search capabilities compared to usual search facilities for learning resources by weighting the dimensions of *skyline queries* [Tan et al., 2001] according to user preferences.

4 Best practices: Examples of successful *architectures* offering personalization in the Semantic Web

In the following sections we present successful *architectures* offering personalization in the Semantic Web.

4.1 Personal Reader

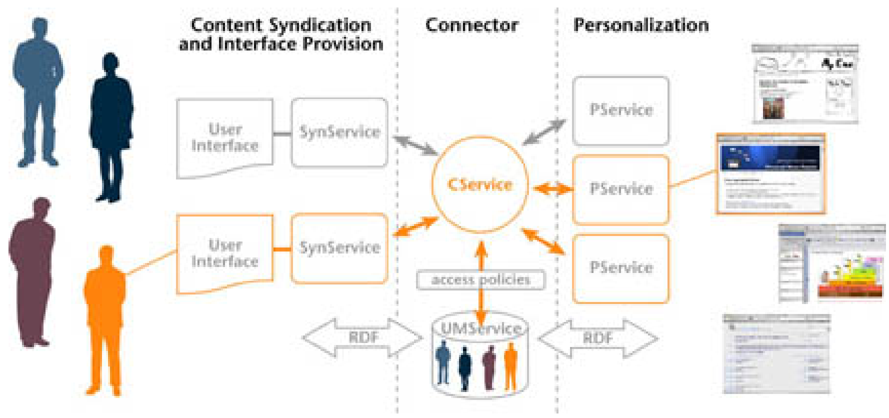

Figure 1: Personal Reader Architecture

The *Personal Reader Framework* [Henze, 2005] enables the creation of modular web service based applications (Figure 1 outlines its architecture). These applications are accessed by device-adaptable user interfaces (UI for short). *Syndication Services* implement the application logic and can be considered as the core of an application. By aid of a *Connector Service*, Syndication Services are able to discover and access *Personalization Services* dynamically, which aggregate domain-specific information in a personalized way. To gather information, Personalization Services access and process Semantic Web data sources. An important feature of the Personal Reader Framework is that new services can be integrated in a *plug-and-play* manner, hence no centralized component has to be modified and new services can be used immediately from all other services within the framework.

Both, Syndication and Personalization Services are able to access and store user data which is supplied by a centralized *User Modeling Service*. The Personal Reader Framework allows to model Syndications Services as statemachines. This facilitates implementation of application logic in multiple ways. Developers just have to implement *action classes* and provide an RDF description of their statemachine, which links those actions with states and events. Moreover, such formal description of application logic eases decoupling the functionality for capturing user observations.

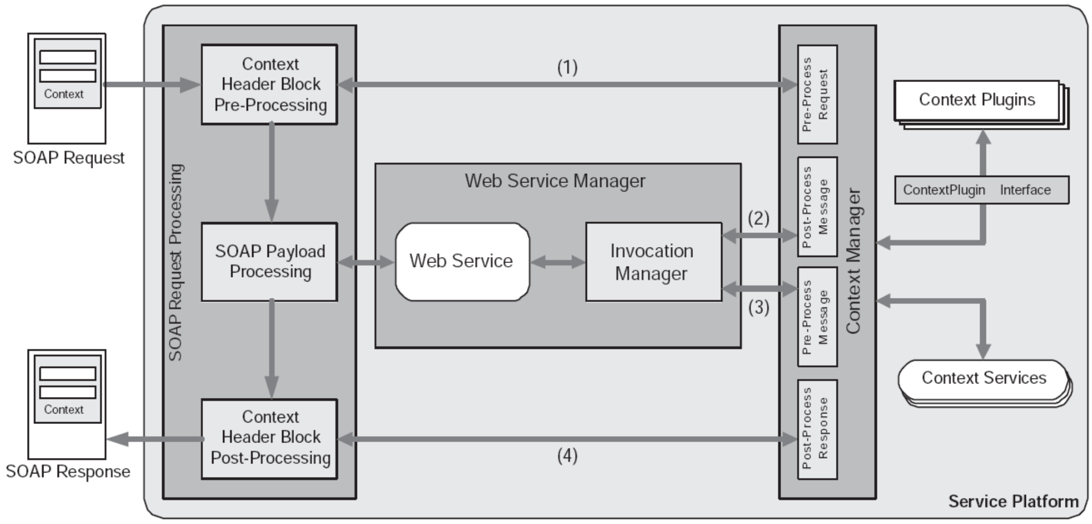

Figure 2: Handling context-enriched web services (source: [Keidl and Kemper, 2004b])

Several applications have been implemented with the Personal Reader Framework like the *Personal Publication Reader* [Baumgartner et al., 2005], the *MyEar* music recommender [Abel et al., 2006] or the *Agent*[4].

In Deliverable A3-D12 we will present a final, enhanced version of the Personal Reader framework in detail.

4.2 Adaptable Web Services

Another architectural solution for enabling personalization in the Semantic Web is described in [Keidl and Kemper, 2004b]. The idea is to deliver context information on invoking web services in order to make such services context-aware and adaptable. Therefore, Keidl and Kemper present a framework facilitates developement and deployment of context-aware adaptable web services and provides an infrastructure that enables communication of context and context processing.

Figure 2 shows the components that are able to process context, which is embedded into the SOAP header. Not only the client (not shown) and the invoked web service are able to utilize context information, but also *context plugins*, which are part of the infrastructure, and *context services*, which are itself web services. Each context plugin and service is associated to a certain context type, e.g. client device characterization, client location, etc. Whenever a web service is invoked – either by a client or a web service – the *Context Manager* enables those context plugins and services, which are associated with type of context embedded in the actual SOAP message, to pre-process the SOAP request. Post-processing of SOAP responses is implemented correspondingly.

MyBook [Keidl and Kemper, 2004a] is an example scenario, in which context plugins and services are used to adapt content provided by amazon web services. Price information is

[4]`http://www.personal-reader.de/Agent`

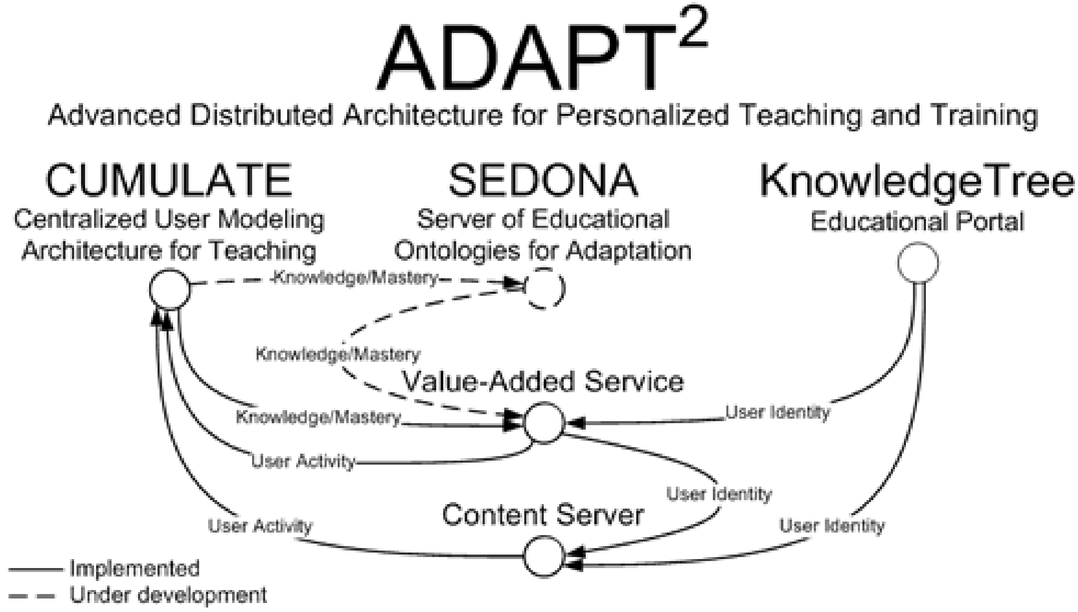

Figure 3: ADAPT^2 overview

converted into currancy at the consumer's current location and context information about the client's device is used to e.g. reject customer reviews if the device type has limitted features.

4.3 ADAPT^2

The *Advanced Distributed Architecture for Personalized Teaching & Training* (ADAPT^2) [Brusilovsky et al., 2005], formerly known as *Knowledge Tree* [Brusilovsky, 2004], aims at providing personalization and adaptation services for developers of otherwise not personalized content. The architecture schema is depicted in Figure 3.

Components of the architecture are:

CUMULATE *CUMULATE* is a cental user modeling server responsible for collecting evidence of user activity and for generating assertions about user knowledge and mastery.

SEDONA *SEDONA* is an "ontology server". It stores assertions about user knowledge and mastery in terms of several ontologies. It is responsible for "translation" of user knowledge between ontologies of the same domain.

Knowledge Tree *Knowledge Tree* is a learning portal that aggregates learning content and structures the courses available for learners. Knowledge Tree authenticates users and maintains user identity records.

Value-Added Service Value-Added Services (VAS) are providing adaptation and personalization. They serve as wrappers for learning objects from content servers. Examples of VAS include *NavEx* and *QuizGuide* [Brusilovsky et al., 2006].

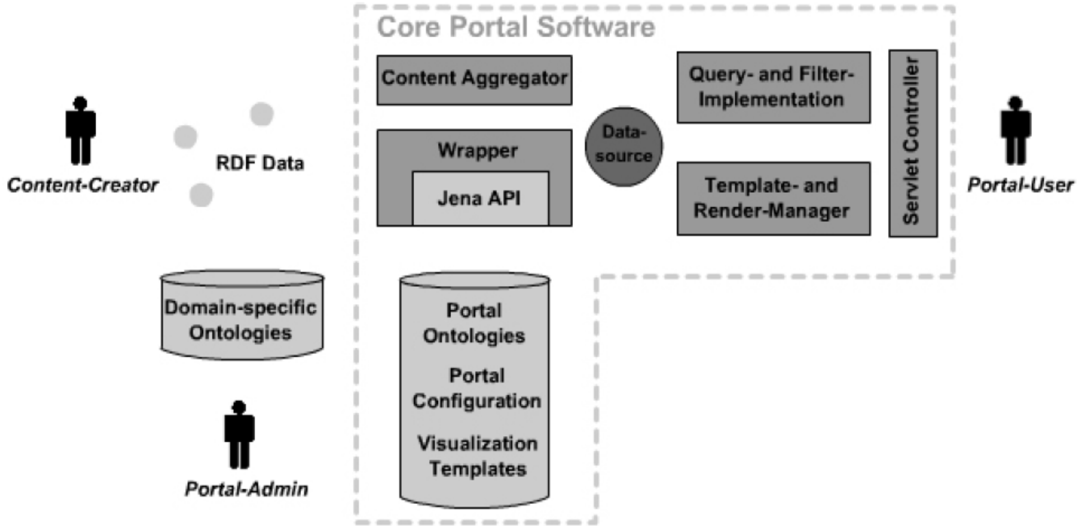

Figure 4: SWAD-E Semantic Portal architecture

Content Server *Content servers* provide learning objects for students to interact with. *WebEx* [Brusilovsky, 2001b] is an example of such content server that work with ADAPT2. WebEx is a Web-based tool for exploring programming examples that enables teachers to use example-based programming approach with heterogeneous classes.

4.4 Personalization in Semantic Portals

Semantic Portals allow for integrated and syndicated data views on information by using ontological knowledge and machine processable semantic descriptions. Figure 4 illustrates the architecture of the *SWAD-E Semantic Portal* [Reynolds et al., 2005].

The *Content Aggregator* is used to read in distributed RDF data utilizing the Jena framework. Aggregated data is encapsulated within a single *Datasource* model, which builds the content base of the portal. The Datasource model encapsulates RDF data, which is aggregated from registered data sources, as well as corresponding schema definitions (*Domain-specific Ontologies*). By use of queries and filters, it is possible to browse the portal data. Rendering of data subsets that match a concrete filter is done by Visualization Templates (*Velocity Templates*).

Navigating through semantic portals is the same as navigating through an RDF graph and is supported with faceted navigation mechanism. Each site a user visits corresponds to an RDF resource and its context that is deduced by aid of domain-specific ontologies. Personalization in semantic portals (cf. Deliverable A3-D5) can be realized in several ways, e.g. personalized filters, personalized preference search (cf. section 3.5), etc. A good strategy is to model users by aid of ontologies that are related to those ontologies that build the vocabulary of the portal's RDF content. Thereby, adaption of content comes almost out of the box as proposed in [Abel and Henze, 2005].

5 Recommendations of working group A3 for personalization in the semantic-based applications and the Semantic Web

5.1 Personalization functionalities

Various personalization functionalities have been researched in more than a decade's research in personalization. Despite the fact that developed techniques have proven their ability to provide user guidance and orientation in hyperspaces, we do not currently see the widespread adoption of these techniques. A couple of reasons may explain this phenomenon. One of them is the current lack of re-usability and interoperability between adaptive techniques/systems, which – to some degree – originates in the so-called 'open corpus problem' [Brusilovsky, 2001a] (cf. cross system personalization[Bhaskar Mehta and Neuhold, 2005]). Thus, one of the main tasks of the work in this working group of the NoE REWERSE was to explore *whether*, and *how* **encapsulated, re-usable personalization** can be realized. Before the start of the project, we observed that personalization functionality was in almost all cases designed from scratch, thus was re-invented for each new application.

We have shown that **encapsulation of personalization functionality** is in principle possible, and have verified the logical description framework that has been developed for this purpose in order to describe personalization functionality on various personalization technologies ([Henze and Nejdl, 2004]). The logical framework separates the core components of the personalization process into independent modules:

Document Space This component maintains and operates on static data. The Document Space contains the hypermedia system in question as well as information associated to this document space. This associated information might be annotations (e.g. metadata attributes, usage attributes, etc.), domain graphs that model the document structure, or knowledge graphs that describe the knowledge contained in the document collections (e.g. domain ontologies).

User Model This component stores, describes and infers information, knowledge, preferences etc. about an individual user (might share some models with the Document Space). The observations (see below) are used for updating the user model.

Observations This component maintains the runtime behavior of the system as far as user interactions are concerned. Examples of observable user interactions include whether a user has visited a document, or visited document for some amount of time, etc. Other examples are rules for compiling e.g. quizzes for testing a user's knowledge on some subject, etc.

Adaptation Component Finally, this component contains the real application logic for personalization, e.g. rules for adaptive functionality (e.g. whether to suggest a document to a certain user, to decide whether to notify a user on interesting news, etc.)

For a thorough discussion, see Deliverables A3-D1 and A3-D4, available at `http://rewerse.net/deliverables/a3-d1.pdf` and `http://rewerse.net/deliverables/m18/a3-d4.pdf`, respectively.

The encapsulation of personalization functionality is one of the main obstacles that needs to be overcome towards the **re-usabilty** of personalized application features. Next to the theoretical achievements with the logical description framework, practical solutions are required that actually *demonstrate* the re-use of personalization functionality. The recommendation of working group A3 is to replicate the encapsulation of personalization functionality by the technic applied for realization: The service technology very well suits this needs, as will be discussed in the subsequent sections.

5.2 Personalization strategies

Personalization of web content can be achieved in various ways as outlined in section 4. When integrating personalization functionalities into applications such functionalities can in general be implemented into two layers: presentation layer and data aggregation layer.

5.2.1 Personalization strategies within the presentation layer

When discussing about personalization within presentation layer an important aspect is organizing content that should actually presented to the user. A helpful approach on organizing content in way that it fits the user's intersts that has become quite popular in the last years in context of Semantic Web applications: faceted navigation [Oren et al., 2006] . Faceted navigation enables users to filter content according to arbitrary factes. For example items of a news feed can be filtered by author, date, subject, etc. There are several browser applications that make use of faceted navigation like *Aduna Spectacle*[5] or *Longwell*[6]. We believe that faceted navigation is preferable to approaches that hide adaptation functionality from the user. And in Deliverable A3-D5, we already present an approach – in context of Semantic Portals – that extends faceted navigation with personalization functionalities.

Adapting user interfaces – e.g. device adaptation, adaptation of style issues, etc. – is another aspect that has to be considered. Therefore, coupling of user interfaces and application logic has to be loose. There exist already a lot of frameworks that assist developers in designing web applications that adhere to corresponding paradigms like *Model View Controller* pattern and there are several technologies like CSS, XSLT, etc. that ease developement of adaptable user interfaces.

5.2.2 Personalization strategies within the data aggregation layers

With the success of the Web 2.0, a new style of web applications has become quite popular: *Mashups.* Mashups partly illustrate a core idea of the Semantic Web which is sharing and reusing web data accross application and community boundaries. Furthermore, they demonstrate what should in our opinion be the web of the near future, namely a *web of services.* Based on a service oriented web, personalization functionality can be applied to both parts, the service providing content and the service that syndicates content from different services.

In order to provide content in a personalized way, services need information about the user's interests and needs. Hence, user profile information has either to be delivered to those services as proposed in section 4.2 and in Deliverable A3-D9, in which we describe so-called *configurable web services* that allow for adjusting (semantic) web servicves to user needs. Or content

[5] http://www.aduna-software.com/products/spectacle/
[6] http://simile.mit.edu/longwell/

providing services must be able to access user profiles. The latter approach requires a component that stores user profile information. We propose that such a component should rather be encapsulated as a separate service as it is demonstrated by the *User Modeling Service* of the Personal Reader framework (cf. Section 4.1) than be integrated within the content providing service. User profiles that are made available as (central) services give the oportunity to let content providers benefit from each other. For example, a service that provides personalized information about music events can benefit from user profile information that was stored by a service that recommends music songs.

In the area of search services, we suggest that *faceted search* is a convenient base for personalization. In [Abel et al., 2007b] we present a strategy that respects user preferences when ranking search results by weighting facets according to those preferences.

Applications that syndicate content from arbitrary sources also benefit from user profiles that are made available as services. On combining content from content providing services, personalization can be applied when calling those services (pre filtering) and when receiving their results (post filtering). Such syndication components have itself already a personal character. For example, creation of *Syndication Services* with the Personal Reader framework (cf. section 4.1) is very easy and thus enables ambitious users with programming skills – e.g. BPEL[7] – to combine *Personalization Services* they like. In turn other users profit from high amount of syndication services because they can use those services too. Web 2.0 tools like *Yahoo pipes*[8] or *QEDwiki*[9] even allow users to build Mashup applications with graphical editors[10].

5.2.3 Required components for architectures facilitating personalization

Summarizing section 5.2.1 and 5.2.2 we propose the following requirements to architectures of web applications that aim for personlized access to web content:

Personalization Services: Services that provide content should be able to do this in a personalized way.

User Modeling Services: User profiles should be shared accross different applications and should implemented as (trustful) services.

Syndication Services: Application logic should be decoupled into personalizable services.

Adaptable user interfaces: User interfaces of syndication services should be adaptable, but should also give users the oportunity to adapt content on their own, as possible with faceted browsing.

Observations captured by user interfaces have to be forwarded to Syndication Services, which for their part can decide how to transform them into user profile data.

[7]http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

[8]http://pipes.yahoo.com/

[9]http://services.alphaworks.ibm.com/qedwiki/

[10]Such tools do not allow for real inter-application integration but illustrate an user-friendly way of how users can be enabled to personalize syndication application of their own.

5.3 Generic Personalization Architecture

According to section 5.2.3 and on the basis of the Personal Reader architecture described in section 4.1 we define a generic architecture for user-adaptable web applications. It is illustrated in Figure 5 and its core components have the following characteristics and features:

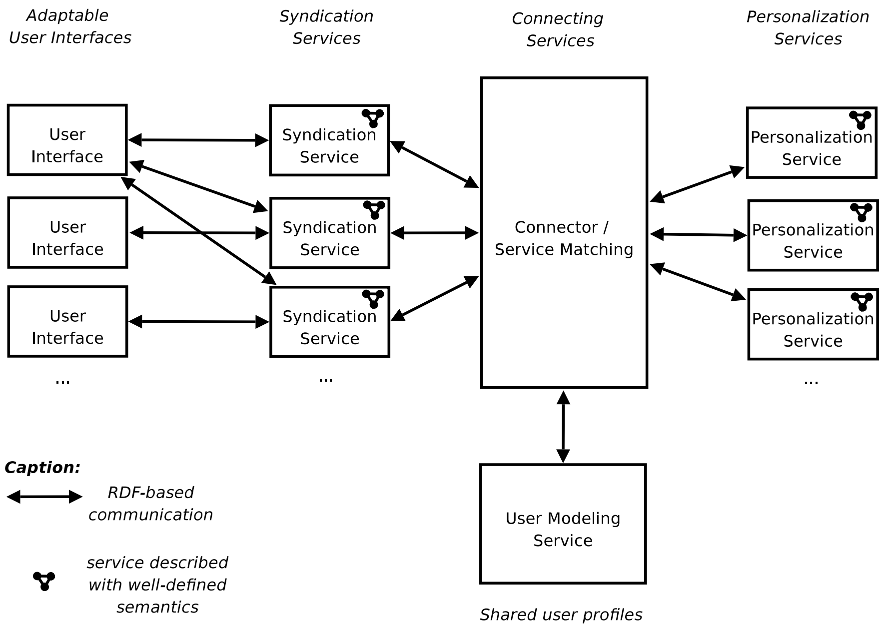

Figure 5: Generic personalization architecture

Personalization Services: Personalization Services are specialized on a certain domain of knowledge. They provide personalized content as RDF. Therefore, they are able to access and store user profile data that is shared accross services. Personalization Services have to be described using well-known standards like OWL-S in order to be discovered and utilized by other services at runtime.

Characteristic of Personalization Services ranges from services that simply wrap non-RDF data sources – e.g. a service that calls the Flickr API considering the user's preferences and transforms the Flickr result into RDF using taxonomies like Dublin Core Metadata Element Set – to services that carry out more complex tasks – e.g. a music recommender service like MyEar (c.f. section 3.4).

Syndication Services: In general Syndication Services implement the application logic. They react to user actions forwarded from a user interface, syndicate content from Personalization Services considering user preferences, may update user profiles and return syndicated RDF content to the invoking user interface. Personalization Services deliver content as RDF. Hence, Syndication Services just have to be aware of the used vocabulary in order to combine content provided by different services.

The Personal Publication Reader (cf. section 3.2) is a good example of a Syndication Service. It makes use of a Personalization Service that provides information about publications and enriches this information with details about athors by utilizing on one hand an *URI resolving service* and on the other hand a service that delivers details about persons.

User Modeling Service: The User Modeling Service manages access to shared user profiles. User profiles are also stored in RDF in order to make these information usable for different services. An User Modelling Service also has to ensure *privacy*. Therefore, the service has to implement access control functionality so that users can decide which service is allowed to add, modify and use what kind of information. In Deliverable A3-D11, we will therefore describe an advanced access control mechanism for RDF stores [Abel et al., 2007a] in detail.

Connector: The Connector component enables dynamic integration of services, thus integration of:

- Personalization Services into Syndication Services
- Personalization Services into another Personalization Service
- User Modeling Service into both, Personalization and Syndication Services

There exist different strategies to match applicable services, ranging from goal-driven approaches – e.g. comparing desired input/output parameters with actual input/output parameters of services – to simple strategies that just map service URIs to actual locations of service decriptions. The latter approach requires basic knowledge about the service that should be integrated.

User Interfaces: An application may offer several user interfaces whereas each user interface may be adapted to a certain device type, e.g. mobile phone, desktop pc, etc. When presenting content user interfaces just visualize RDF data. For that part it is possible to use generic user interfaces suitable for different Syndication Services, e.g. RDF browsers that feature faceted navigation. Furthermore user interfaces have to enable users to formulate requests. Even this part can be covered by generic user interfaces if Syndication Services have a semantic description of their features, e.g. an OWL-S description. When implementing user interfaces developers may utilize common web frameworks like Velocity[11] or Ajax frameworks like Echo2[12].

In general communication between components of the architecture should be RDF-based so that cooperation between components not knowing each other is simplified. And at least those services that are plugged together should be described using well-defined semantics like OWL-S.

[11] http://velocity.apache.org/
[12] http://www.nextapp.com/platform/echo2/echo/

6 Conclusions

This report summarizes the findings and recommendations of the working group A3 on personalized information systems for establishing personalization in semantically enhanced environments and the Semantic Web. After a brief discussion on the relation between semantic technologies and their benefits for personalization, a synoptical overview on *examples of successfully realized personalized information systems* and *promising architectures* is given. The report ends with stated recommendations for personalization in the Semantic Web, separated into the dimensions of *personalization functionality*, *personalization strategies*, and *enabling architectures.* A more in-depth discussion on the systems and architectures developed within REWERSE and discussed in sections 3 and 4 will be given in Deliverable A3-D11.

One final recommendation of the working group concerns *good practices* in personalization, and is valid not only in the context of Semantic Web but whatsoever attempt to realize personalized access to digital information. Personalization and usability are highly intertwined, and *personalization is no cure-all* for badly designed systems. In the contrary, lots of improvement can be done by improving usability, and by analyzing which user groups need to be served, what are required user interaction processes, and where can personalized shortcuts / personalized navigation aid / personalized information presentation / etc. *further* improve the usefulness of the system.

Acknowledgement

This research has been co-funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. `http://rewerse.net`).

References

[Abel et al., 2006] Abel, F., Brunkhorst, I., Henze, N., Krause, D., Mushtaq, K., Nasirifar, P., and Tomaschweski, K. (2006). Personal reader agent: Personalized access to configurable web services. Technical report, Distributed Systems Institute, Semantic Web Group, University of Hannover.

[Abel et al., 2007a] Abel, F., De Coi, J. L., Henze, N., Koesling, A. W., Krause, D., and Olmedilla, D. (2007a). Enabling advanced and context-dependent access control in RDF stores. In Aberer, K., Choi, K.-S., and et al., N. N., editors, *International Semantic Web Conference 2007 (ISWC 2007) (to appear)*, Busan, Korea.

[Abel and Henze, 2005] Abel, F. and Henze, N. (2005). User Awareness and Personalization in Semantic Portals. In *Proceedings of 4th International Semantic Web Conference, Galway, Ireland (6th–10th November 2005).* Digital Enterprise Research Institute.

[Abel et al., 2007b] Abel, F., Herder, E., Kaerger, P., Olmedilla, D., and Siberski, W. (2007b). Exploiting preference queries for searching learning resources. In *in proceedings of the 2nd European Conference on Technology Enhanced Learning (EC-TEL 2007).*

[Baldoni et al., 2006] Baldoni, M., Baroglio, C., Brunkhorst, I., Henze, N., Marengo, E., and Patti, V. (2006). A Personalization Service for Curriculum Planning. In Herder, E. and

Heckmann, D., editors, *Proc. of the 14th Workshop on Adaptivity and User Modeling in Interactive Systems, ABIS 2006*, pages 17–20, Hildesheim, Germany.

[Baldoni and Marengo, 2007] Baldoni, M. and Marengo, E. (2007). Curriculum Model Checking: Declarative Representation and Verification of Properties. In Duval, E. and Klamma, R., editors, *Proc. of EC-TEL 2007 - Second European Conference on Technology Enhanced Learning*, LNCS. Springer.

[Baumgartner et al., 2005] Baumgartner, R., Henze, N., and Herzog, M. (2005). The personal publication reader: Illustrating web data extraction, personalization and reasoning for the semantic web. In *ESWC*, pages 515–530.

[Bhaskar Mehta and Neuhold, 2005] Bhaskar Mehta, Claudia Niederee, A. S. M. H. and Neuhold, E. (2005). Towards cross-system personalization. In *UAHCI*.

[Brusilovsky, 2001a] Brusilovsky, P. (2001a). Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110.

[Brusilovsky, 2001b] Brusilovsky, P. (2001b). Webex: Learning from examples in a programming course. In *WebNet'2001, World Conference of the WWW and Internet*, pages 124–129, Orlando, FL, USA.

[Brusilovsky, 2004] Brusilovsky, P. (2004). Knowledgetree: a distributed architecture for adaptive e-learning. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 104–113, New York, NY, USA. ACM Press.

[Brusilovsky et al., 2005] Brusilovsky, P., Sosnovsky, S., and Yudelson, M. (2005). Ontology-based framework for user model interoperability in distributed learning environments. In *E-Learn 2005*.

[Brusilovsky et al., 2006] Brusilovsky, P., Sosnovsky, S., and Yudelson, M. (2006). *Addictive Links: The Motivational Value of Adaptive Link Annotation in Educational Hypermedia*.

[Henze, 2005] Henze, N. (2005). Personalization services for the semantic web: The personal reader framework. In *Framework 6 Project Collaboration for the Future Semantic Web Workshop at European Semantic Web Conference ESWC 2005*, Heraklion, Greece.

[Henze and Nejdl, 2004] Henze, N. and Nejdl, W. (2004). A Logical Characterization of Adaptive Educational Hypermedia. *New Review of Hypermedia*, 10(1).

[Keidl and Kemper, 2004a] Keidl, M. and Kemper, A. (2004a). A framework for context-aware adaptable web services. In *EDBT*, pages 826–829.

[Keidl and Kemper, 2004b] Keidl, M. and Kemper, A. (2004b). Towards context-aware adaptable web services. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 55–65, New York, NY, USA. ACM Press.

[Kłopotek et al., 2007] Kłopotek, M. A., Wierzchoń, S. T., Ciesielski, K., Dramiński, M., and Czerski, D. (2007). Conceptual maps of document collections in internet and intranet. *Coping with Technological Challenge. IPI PAN Publishing House.*

[Müller et al., 2001] Müller, C., Großmann-Hutter, B., Jameson, A., Rummer, R., and Wittig, F. (2001). Recognizing time pressure and cognitive load on the basis of speech: An experimental study. In *User Modeling 2001: 8th International Conference, UM 2001*, Sonthofen, Germany.

[Oren et al., 2006] Oren, E., Delbru, R., and Decker, S. (2006). *Extending Faceted Navigation for RDF Data*.

[Reynolds et al., 2005] Reynolds, D., Shabajee, P., Cayzer, S., and Steer, D. (2005). Semantic portals demonstrator - lessons learnt. Technical Report Deliverable 12.1.7, SWAD-Europe project.

[Tan et al., 2001] Tan, K.-L., Eng, P.-K., and Ooi, B. C. (2001). Efficient progressive skyline computation. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 301–310, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.