

A Rule-based Adaption Model for Ontology-based Personalization

Thanh Tran, Philipp Cimiano, Anupriya Ankolekar

Institute AIFB, University of Karlsruhe, Germany
{dtr, pci, aan}@aifb.uni-karlsruhe.de

Abstract. Various adaptive hypermedia systems have been proposed to alleviate information overload on the Web by personalising the delivery of information and resources to the user. These systems have however been afflicted with difficulties in the acquisition of an accurate user model, a limited degree of customization offered to the user as well as general lack of user control on and transparency of the systems' adaptive behavior. In this paper, we argue that the use of rules on top on ontologies can enable adaptive functionality that is transparent and controllable for users. To this end, we present ODAS, a domain ontology for adaptive hypermedia systems, and a model for the specification of ODAS-based adaptation rules. We demonstrate the use of this model by showing how it can be instantiated within a knowledge portal to arrive at rules that exploit ODAS semantics to perform meaningful personalization.

1 Introduction

Without assistance, the vast information space of the Web leads to confusion for the average user, manifested by "comprehension and orientation problems" and a general "loss in information space"[19]. A common approach to reduce such confusion, used by many existing Web portals, such as Yahoo¹, Ebay² and Amazon³, is to filter out irrelevant information for the user and only provide personalized content. These portals are often both **adaptable**, i.e. allow for manual configuration by the user, as well as **adaptive**, i.e. providing proactive personalization to address the ad-hoc needs of the user. Such adaptive behaviour is typically realised by either collaborative filtering [15], which identifies content found relevant by similar users, or content-based filtering [16], which exploits similar content to identify relevant resources for a certain user.

In order to provide better personalization functionality, more dimensions such as the task, the system and the environment need to be considered. The increase in contextual information and the use of advanced machine learning techniques leading to better user models can be expected to provide improved personalization functionality. However, apart from the inherent difficulties in collecting model information, these approaches are criticized as being black-boxes that cannot be inspected.

¹ <http://www.yahoo.com>

² <http://www.ebay.com>

³ <http://www.amazon.com>

A major drawback of both filtering approaches is that the models they develop of users and content are black-box-like and not amenable to user inspection or modification. If, however, we could represent the adaptation logic in the form of rules, users would have much greater control over the system. They would be able to inspect, possibly understand, and even modify the underlying adaptation model. In this paper, we present a model for the specification of such adaptation rules. In addition, we present several examples to illustrate how this rule-based model can be instantiated to specify different styles of adaptive behavior.

In this paper, we argue that an ontology-based approach to personalization can successfully address these drawbacks. We discuss an ontology capable of capturing information about the user, the task, the system, the environment, and various aspects of the content (structure and presentation), i.e. all dimensions that have been proven to be useful in existing approaches. This maximizes the amount of contextual information that can be used to accomplish sophisticated adaptation. Moreover, current adaptive hypermedia systems rely on their own formalism and vocabulary for data representation. By the use of a standardized ontology, we argue that systems can share and reuse model information to solve the inherent lack of data that hinders sophisticated personalization.

In the remainder of the paper, we first discuss in greater detail the benefits of an ontology-based personalization approach in Section 2. Then, in Section 2.3, we briefly present a domain ontology, ODAS, to introduce some concepts that we will use in the specification of adaptation rules. The main bulk of this paper is contained in Section 3, which introduces and illustrates by way of examples, a rule-based model for ontology-based adaptation. Finally, we review other ontology-based approaches in Section 5, before pointing towards future work in Section 6.

2 Overview of Adaptation Approaches

In this section, we present a brief survey of the main adaptation approaches and their drawbacks, arguing that ontology-based approaches plus the use of rules can overcome most of these drawbacks.

2.1 State-of-the-art

Many authors agree on three different levels of adaptation. Whereas **adaptive content** focuses on selecting the appropriate content in different multimedia resources, **adaptive presentation** is concerned with the visualization of the content, the interactive elements and the appropriate media. Finally, **adaptive structure** deals with the number, appearance, order and target of links [2].

In current commercial applications, filtering-based approaches are state-of-the-art. Content-based filtering applies different similarity measures on a representational model of the content. So, if the user prefers a product, similar product pages will be suggested [16]. As the system always recommends more of what the user has already indicated an interest for, there is a potential problem of **overspecialization**, reducing the possibility of serendipitous finds.

Collaborative filtering assumes similar users have similar interests, so recommendations are based on user similarity [15]. This assumption reduces the role of individual

preferences. In addition, it is not possible to recommend a **new item** not yet rated by users. When there is only **sparse data**, i.e. when the number of users is small relative to the amount of content, typically only a small percentage of the resources will be accessed by the user. The sparse coverage of ratings then results in an accordingly sparse set of recommendable items [20].

Noting that both approaches are complementary, some authors suggest combining them [1]. Subjective ratings of user groups counterbalance the lack of subjective data about resources. New items can be recommended on the basis of their content. Also, sparsity is solved when recommendations do not depend merely on user ratings. Yet, incorporating even more sources can further improve personalization. It has been shown that information related to the task [3], the environment and the system [13] can be relevant for adaptation. Furthermore, by means of an explicit presentation model, an adapted presentation of the content can even be generated on the fly (compare [23]).

2.2 On the Use of Ontologies and Rules for Adaptation

While these advances have led to sophisticated adaptive hypermedia systems, there are still two drawbacks that we think can be addressed by the use of ontologies and rules.

First of all, these systems are seen as **black boxes**, computerized oracles which give advice but do not make their model explicit such that it can be inspected [11]. This is due to the fact that the underlying algorithms compute latent factors and heuristics that cannot be translated to rationales for the adaptive behavior. Consequently, the reasons behind the recommendations cannot be explained to the user. This is different when using rules. When adaptive behavior is captured by rules, inference engines processing them produce recommendations that are more accessible to the user. Underlying inferences can be analyzed, provided with a proof and the rules can be made available for the user to be inspected and modified. This could allow for feedback loops, user-control and thereby enhance user trust towards the system. These benefits will be illustrated in section 3.

Second, current adaptive hypermedia systems face difficulty in **obtaining model information**. Sophisticated user models are hard to develop in systems where the duration and the type of interactions do not provide much relevant information. Yet users may expose lots of useful information at applications they frequently use. Note that the quality of adaptation depends on the amount of information that can be drawn on. Consequently, adaptive hypermedia systems benefit from the ability to exchange and reuse information. This ability is particularly important in the context of open-corpus adaptation [9]. Such a corpus may be the Web and the ultimate objective is to reduce the users' cognitive overload by providing a personalized view on the resources available on the Web.

Currently, the exchange of information is made possible mostly by using XML-based protocols. An XML schema defines the syntax, i.e. the structure and labels of the data elements that are to be exchanged across systems. In this respect, it may be sufficient to exchange standard metadata and model information that are known to the involved parties, e.g. the title and author information of a PDF document. They have implicitly agreed on the semantics of these data elements and a priori know how to process them. However, exchanging data in the domain of adaptive hypermedia systems,

and especially in an open-corpus, requires more flexibility. In fact, adaptive behavior may build on explicit, semantic descriptions of the current task of the user, the resources he/she is currently reading, etc. in form of an ontology. Thus, the semantics of the information transferred and reused across systems must be formalized so that it can be exploited in the adaptation process. For instance, the semantics of the content syndicated from external sites must be processable to the user's system in order for them to be personalized in a dynamic and straightforward manner.

Ontologies can enable such semantic interoperability. Adding further capabilities to the stack already available with XML, ontologies can be used to specify the semantics of data elements shared across systems. With an XML schema, the semantics is only available to the people that have specified it. Using ontologies, the semantics can be determined automatically by the systems at runtime [22]. Therefore, the use of ontologies can address the lack of data in current adaptive hypermedia systems. They can share and reuse data, and on the basis of the semantics formally specified in the accompanied ontologies, they can make sense of the data to exploit it for adaptation. In the same manner, ontologies can facilitate adaptation in an open corpus where the semantics of the available resources is made explicit to the systems.

2.3 ODAS - A Domain Ontology for Adaptive Hypermedia Systems

A crucial point in the development of adaptive hypermedia systems is the identification of information sources other than related to the user and content. Each of these sources establishes one aspect of the adaptation context that can help to improve the sophistication and effectiveness of adaptive behavior.

Fig. 1 shows a portion of the subclass hierarchy of ODAS, an ontology for the domain of adaptive hypermedia systems we have developed. We will now illustrate how the different aspects of the context can be represented by the use of this ontology and why they are useful for adaptation. The different adaptation dimensions are highlighted by rectangles in Fig. 1 and, henceforth, will be referred to as models.

Central to the representation of the adaptation context is the notion of `Process`. `Application Interaction`, for instance, tells the system that a particular `User` is currently interacting with a `Content` resource (**resource model**) of the `Application` (**system model** part of the **environment model**) to accomplish a task. Indirectly, this task may have been modelled in the system as a `Composite Process`. Precisely, the workflows required to accomplish the tasks supported by the system can be represented in the system as instances of a `Computer-aided Process` (**task model**). Such a process-orientated representation of a task comprises many `Application Interactions`. When the current interaction is identified as part of such a workflow, the modelled output of the respective `Process` can be implicitly assumed to be the user's goal. Ideally, the adaptation should then be performed in a way that supports the user in achieving this goal. Section 3.3 shows rules that make use of the relations among sub-activities of this task model, e.g. `is pre` and `is post`, to perform task-based adaptation.

`Application Interactions` thus contain information about the `Content` currently processed by the `User`. Different content types are distinguished by the subjects they describe. While `Content about Entity` has exactly one ontology en-

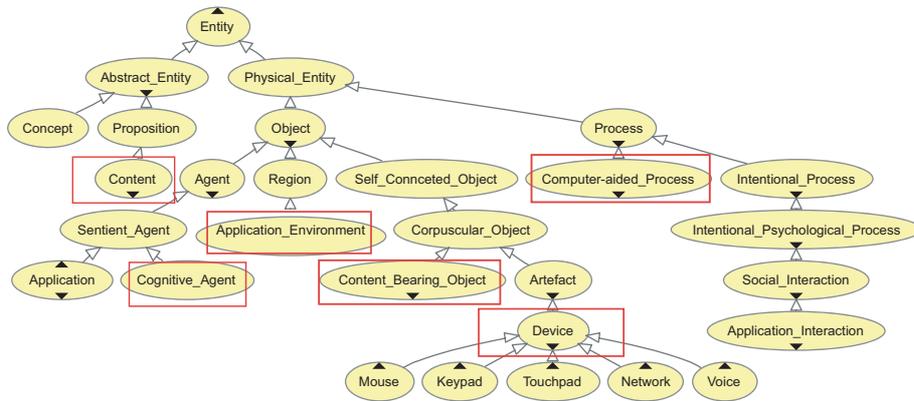


Fig. 1. ODAS Concept Hierarchy

tity as subject, Content about Relation has several such subjects. The subject is in fact the "content semantics" that ODAS aims to capture. Section 3.3 contains several content-based adaptation rules that exploit this notion of semantics to trigger recommendations representing different types of Content resources related to the one the user is currently interacting with.

Interesting is also the Executable Content concept, which differs from other types in that it is embodied in a UI Element and represents a Service. This is useful because in addition to information resources, services can also be considered in adaptation. In fact, any services can be associated with an individual of Executable Content and be adapted to the User just like other content types. Furthermore, the concept of Content Bearing Object (CBO) allows for a distinction of the actual materialization from the abstract Content embodied in it. Different variations of layout- and presentation of the same resource can be captured by an instance of CBO and the resulting **presentation model** supports the concept of adaptive presentation.

Further concepts that deliver contextual information are User and Environment. User properties such as has credential, has read, knows, has interest or is able to can be used to reflect constraints that need to be considered (**user model**). Also, characteristics of the Environment can play a similar role in adaptation (**environment model**). Restrictions given in the Environmental Components such as bandwidth of the Network and size and resolution of the Display Device can be incorporated to only deliver resources which comply with the specified constraints.

3 A Rule-based Adaptation Model

While the ontology represents the different adaptivity dimensions in terms of user, resource, task, environment and system model, this section demonstrates that the logic underlying the system behavior can also be explicitly captured on the basis of a rule-based adaptation model. This section begins with an illustration of the main ideas on the

basis of a personalized portal. Then, we present a formalization of the abstract adaptation model as well as a concrete instantiation based on DL-safe rules. Finally, the different part of the rules and different possible instantiations of the adaptation model will be discussed.

3.1 Applying Rules-based Adaptation to a Portal

Figure 2 shows a personalized portal, a prototype implementation that shall demonstrate the main ideas of the approach proposed here. The adaptive part of the portal is an extension of the Liferay architecture framework that encompasses four functional modules. Apart from basic navigation and search, the system is able to track user interactions, generate recommendations in the form of links and apply different layouts to the content chosen by the user.

The presentation module in Figure 2 shows a user reading "Introduction" – a section of a `Composite Content` unit, i.e. a scientific paper, with the title "From SHIQ and RDF to OWL...". This establishes a context that the system can exploit for adaptation. The result of adaptation is a list of recommendations, i.e. context-relevant resources that are presented by the recommendation module as hyperlinks. As shown, resources assessed by the system as relevant to the context also have OWL as subject – and additional ontology entities such as `OWL semantics` and `RDF` as shown in brackets. Though only content-related information is exploited by the system in its current implementation, in principle the approach also supports adaptation dimensions related to structure and presentation. In fact, the actual context model is more complex and involves many of the models discussed previously.

Firstly, such recommendations based on the context semantics are possible by the annotation of each paragraph with the entities that it deals with. In order to account for this so called resource model, object properties of `Content` such as `subject` are used. Different layout variations of the contents are represented via the `CBO` concept and serve as presentation models. In the future, the context employed by the system will be extended with further dimensions. System- and environment-related information such as bandwidth, resolution and other properties of the client device will be taken into account. Technically, precise metrics are assessable on a desktop application, but with the portal system it may suffice (and is more feasible) to obtain the client device's properties as nominal values only. Also, task related information will be incorporated. An expert can model a flow of activities involving dependent content units (and services). For instance, booking a flight (represented as `Computer-aided Process`) can be modelled as consisting of a sequence of atomic interactions (represented as `Executing`, subclass of `Application Interaction`) such as destination selection, car booking and hotel booking. Using this information, the system can recommend the `Executable Content` units associated with these services to help the user accomplishing the task of flight booking.

Most crucially, the interactions between the user and the system are recorded. For making recommendations, the system needs to know what the user is currently reading, what service is being invoked, for how long, with what result etc. This establishes the actual context and allows the system to update the user model. In fact, relevant properties of the user which change the course of the interaction are propagated to the

Welcome, thanh tran!
Home - My Account - Sign Out

Adaptation Business Personal

Content and Layout Look and Feel My Communities Desktop

Search

Content [Go]

Navigation

Adaptation

Wiki

Private Nodes
Public Nodes

Weather

90210	71.0 °F	
Chicago	35.0 °F	
Frankfurt/Main	37.0 °F	
Rome, Italy	46.0 °F	
Beijing	21.0 °F	

City or Zip Code

Search

Admin

Company	Users
Portlets	Groups
Server	Roles

Adapted Recommendations

Content related to "Introduction (about: OWL)"

- Avoiding Paradoxes (about: OWL , OWL Semantic)
- Future extensions (about: OWL , Evolution of OWL)
- Handling Malformed Graphs (about: OWL , OWL Syntax, OWL Semantic)
- OWL as a Description Logic (about: OWL , DL Class)
- OWL Full as an RDF Extension (about: OWL , RDF)
- Influences on OWL (about: OWL , DL class, Frames Model, RDF syntax)
- Providing a Viable Semantic Theory for OWL (about: OWL , OWL Semantic)
- Readability (about: OWL , OWL Semantic)
- Retaining Decidability (about: OWL , Reasoning with OWL)
- Semantics for OWL DL (about: OWL , OWL Semantic)
- Semantics for OWL Full (about: OWL , OWL Semantic)
- Predecessors of OWL (about: OWL , SHOE, DAML-ONT, OIL, DAML+OIL)

Recommendations

Content similar to "Introduction (about: OWL)"

Illustration for "Introduction (about: OWL)"

Service for "Introduction (about: OWL)"

Adaptive Content

+ From SHIQ and RDF to OWL: The Making of a Web Ontology Language

Introduction

Content currently read

- - paragraph 1

OWL [10] is a new ontology language for the Semantic Web, developed by the World Wide Web Consortium (W3C) Web Ontology Working Group. OWL was primarily designed to represent information about categories of objects and how objects are interrelated—the sort of information that is often called an ontology. OWL can also represent information about the objects themselves—the sort of information that is often thought of as data.

- + paragraph 2

Fig.2. An ODAS-based Personalized Portal

knowledge base (KB). For instance, suppose the user chooses one of the recommendations shown in Figure 2. Then, if there is a timeout greater than the time amount estimated to be the `Content`'s `minimum processing time`, the system assumes that the user `has read` the content and will update the user model accordingly. And when the user has read the same content for a number of times, the system assumes the he/she then `knows` it.

Thus, there are many contextual dimensions that are relevant for adaptation and ODAS is used to represent them. The entire context being captured can then be regarded as a list of conditions. Given a relevant list, the system will generate the most appropriate recommendations. In other words, when a situation is recognized by the system, it will adapt the resources accordingly. Modelling this behavior in the form of rules is indeed the core of the approach presented here. The logic for adaptation is specified declaratively in the form of rules representing some generic if-then patterns. Such patterns, i.e. a logic program, is at a higher conceptual level than if-then statements in a purely imperative program.

In particular, some generic adaptation contexts are declaratively specified as a set of conditions part of adaptation rules that, when satisfied, lead to the inference that some resource is recommended. Consequently, the procedural code needed for the recommendation module is straightforward. It comprises of only a few retrieval operations. Most of the work is done by the inference engine. For the portal, we use KAON2 as knowledge backend [14]. It supports the persistent storage of the many employed models, i.e. ODAS concepts and instances. A modelling service continuously inserts new instances of `Application Interaction` and thus triggers the processing of DL-safe adaptation rules. Such rules are constrained in a way that allows for integration with Description Logics – OWL in particular – while keeping reasoning decidable (see [18]). They can be processed efficiently with the KAON2 inference engine. This engine checks if the inserted interactions match the conditions stated in the rules and fires them accordingly.

3.2 A Formal Definition of the Adaptation Model

We will begin with a formalization of our proposed adaptation model. Let CC be the set of all context conditions, AC the set of adaptation conditions, UC the set of user conditions, EC the set of environment conditions and R the set of recommendations; CC , AC , UC , EC and R are pairwise disjoint; let $V = (EC \cup UC \cup AC \cup CC) \cup R$ be the set of all conditions and recommendations. An adaptation model then is defined as a finite set of adaptation rules of the following form, where r_i^+ are positive head atoms, ec_i^+ , uc_i^+ , ac_i^+ , cc_i^+ are positive body atoms and $\neg ec_1^-$, $\neg uc_1^-$, $\neg ac_1^-$ and $\neg cc_1^-$ are negative body atoms:

$$(r_1^+ \oplus \dots \oplus r_n^+) \leftarrow \begin{array}{l} [ec_1^+ \oplus \dots \oplus ec_m^+ \wedge \neg ec_1^- \oplus \dots \oplus \neg ec_l^-] \wedge \\ [uc_1^+ \oplus \dots \oplus uc_k^+ \wedge \neg uc_1^- \oplus \dots \oplus \neg uc_x^-] \wedge \\ [ac_1^+ \oplus \dots \oplus ac_x^+ \wedge \neg ac_1^- \oplus \dots \oplus \neg ac_z^-] \wedge \\ [cc_1^+ \oplus \dots \oplus cc_v^+ \wedge \neg cc_1^- \oplus \dots \oplus \neg cc_w^-] \end{array} \quad (1)$$

where $\oplus \in \{\vee, \wedge\}$.

These atoms stand for the conditions and recommendations captured in the rules, i.e. they are elements of V , more precisely, $ec_i \in EC$, $uc_i \in UC$, $ac_i \in AC$, $cc_i \in CC$ and $r_i \in R$. Hence, the rule head is a conjunction of recommendations expressed in the form of logical formulas. The rule body is a conjunction of elements in EC , UC , AC and CC , each consisting of an arbitrary combination of disjunctions and conjunctions of positive and negative conditions also expressed as logical formulas.

Note that this adaptation model is formalized in a way independent of a logical formalism. Thus, the semantics depends on the concrete formalism that is used to formulate a particular adaptation model. We now continue to illustrate the application of this abstract model on the basis of an instantiation expressed as DL-safe rules.

The DL-safe rule fragment constitutes a decidable combination of OWL-DL and function-free Horn rules. Such a rule has the form $H \leftarrow B_1, \dots, B_i$ where H and B_i are atoms; H is the rule head, and the set of all B_i constitute the rule body. An atom has the form $A(t_1, \dots, t_n)$ where A is a predicate and t_i are terms; a term is either a constant or a variable. A rule r is called DL-safe if each variable in r occurs in a non-DL atom in the rule body where a DL-atom is $A(s) \in N_C$ or $R(s, t) \in N_{R_a} \cup N_{R_c}$ and $N_c \cup N_{R_a} \cup N_{R_c} \subseteq N_p$, the set of predicate symbols of a description logic knowledge base. A DL-safe rule-based adaptation model is a finite set of such rules, i.e. a logic program P containing only of DL-safe rules. The semantics of a DL-safe rule is defined by the equivalence to the clause $H \vee \neg B_1 \vee \dots \vee \neg B_n$ [18].

Note that as opposed to the abstract model, the expressivity of this DL-safe instantiation is more limited, e.g. rules are not allowed to contain negation or disjunctions in the rule body. Nevertheless, the DL-safe rule fragment has practical relevance. Using this formalism, statements expressed in OWL-DL can be incorporated into adaptation rules and be more or less efficiently processed by inference engines. The following adaptation rule 2 illustrates how knowledge represented with ODAS can be exploited for the purpose of formulating adaptation rules. We will now discuss this and other possible instantiations of the adaptation model using ODAS.

$$\begin{aligned}
needs(u, z) \leftarrow & [Reading(p), CognitiveAgent(u), agent(p, u), \\
& ContentAboutEntity(c), resource(p, c), \\
& [ContentAboutRelation(y), CBO(z), Entity(e), \\
& hasSubject(c, e), hasSubject(z, e), contain(z, y),] \quad (2) \\
& [Credential(h), requiresCredential(z, h), \\
& hasCredential(u, h), interestIn(u, z),] \\
& [Voice(v), instrument(p, v), Audio(z)]
\end{aligned}$$

3.3 ODAS-based Instantiations of the Adaptation Model

As shown in rule 2, ODAS concepts and properties are used to express conditions on recommendations. Context-related conditions are captured by the notion of `Process` and its relations to other ontology entities. This concept serves as the "entry point" to access various models, i.e. `Content`, `Task`, `User` and `Environment`. Rule 2 shows that the `Content` concept can be used to express conditions of the adaptation part. In particular, the adaptation logic specified there is based on subconcepts of `Content`, i.e.

Content about Relation and Content about Entity. While this is referred to as content-based adaptation, we will also discuss other possible instantiations of this part that lead to different adaptive behaviors. Eventually, these different styles of adaptation yield a set of resources related to the one the user is currently interacting with. In the last two parts, properties related to the `User` and the `Environment` act as constraints that, when applied, have a minimizing effect on this adapted set of resources. This ensures that the resources are indeed suitable to the user and the environment. Note that this is due to the fact that rules fire only when all conditions in all parts are satisfied. We will now elaborate on the instantiations of these parts of the rule in more details.

Content-based Adaptation – Semantically-related contents In ODAS, the (semantic) content of a resource is described by the property `has_subject`. In essence, it tells which ontology entities a content unit deals with. Based on this representation, resources can be considered as related if they have the same entity or related entities as subjects. In this regard, entities can be considered related if they are directly or indirectly connected – through some other entities – via some properties specified in the ontology. Exploiting the ODAS conceptualization, the adaptation-related part of the rule may consist of the following conditions:

$$\leftarrow [\dots] \text{Entity}(z), \text{Content}(x), \text{hasSubject}(x, z), \quad (3)$$

$$\text{Content}(y), \text{hasSubject}(y, z) [\dots]$$

This would lead to the recommendations of all content units y which are related to the content x currently processed by the user via the same entity z they have as subject – or to be precise, any entities that can be substituted for the variable z . The following are two other instantiations of the adaptation part of the rule. In a similar manner, they ensure that recommendations encompass related content units:

$$\leftarrow [\dots] \text{Entity}(u), \text{Entity}(v), \text{isRelatedto}(u, v), \text{Content}(x), \quad (4)$$

$$\text{hasSubject}(x, u), \text{Content}(y), \text{hasSubject}(y, v) [\dots]$$

$$\leftarrow [\dots] \text{Entity}(u), \text{Entity}(v), \text{Process}(p), \text{isInvolvedIn}(u, p), \quad (5)$$

$$\text{isInvolvedIn}(v, p), \text{Content}(x), \text{hasSubject}(x, u),$$

$$\text{Content}(y), \text{hasSubject}(y, v) [\dots]$$

As opposed to 4, contents recommended by the examples 5 and 6 are related to the current content not because they describe the same but a related entity. In 5, entities that can be bound to u and v are involved in a particular relationship. Precisely, they are connected by `is related to`, or by any sub-properties thereof, respectively. Note that in ODAS, this is the top property, i.e. the super-property of any object properties defined. When the integration of domain ontologies into ODAS respect this semantics,

i.e. domain properties are also mapped as sub-properties of `is related to`, then any pairwise related entities can be substituted for u and v . In 6, entities bound to u and v are indirectly related because they participate in the same `Process`. Since `agent`, `target` etc. are sub-properties of `is involved in`, there may be many entity types that are related in this way.

In fact, these instantiations of the adaptation model produce the results shown in the portal in Figure 2. As the user is reading "Introduction" (OWL), which is a `Content` about `Entity` describing OWL (entities in brackets stand for the subjects), "Avoiding Paradoxes" (OWL, OWL Semantics) is one of the recommendations generated by the portal as a result of processing the instantiation in 4, "Predecessors of OWL" (OWL, SHOE, DAML-ONT, OIL, DAML+OIL) as a result of adaption rule 5 and "Future extensions" (OWL, Development of OWL) as a result of the adaption rule 6.

It is worth noting that we assume that this adaptation model mimics in some way the style of authors who start with a section describing a key entity, e.g. OWL, in a general, introductory way. In subsequent sections, the author goes into details by focusing on some of its relationships to other entities that are relevant from his/her point of view, e.g. relationships among OWL, SHOE, OIL etc. The counterpart to this narrative style is to start with an overview of a complex phenomenon involving many entities and proceed with subsections, each focusing on one of these entities. Using ODAS `Content` about `Relation` and a related `Content` about `Entity`, this may be captured in the adaptation part as follows:

$$\leftarrow [\dots] \textit{Entity}(u), \textit{ContentAboutRelation}(x), \textit{hasPrimarySubject}(x, u), (6) \\ \textit{ContentAboutEntity}(y), \textit{hasSubject}(y, u)[\dots]$$

The concept `Content` about `Relation` is used to describe (and can be exploited by the reasoner to classify) instances having more than one entity as subjects. The aim is to reflect the semantics of content units that deal with a complex phenomenon – rather than one single entity like instances of `Content` about `Entity` do. Such a phenomenon may be a relationship, i.e. one-, two-, n-ary relations between entities that may reflect a causality, a correlation, an interrelation or a connection in general. Due to the limitation in the expressiveness of the language underlying ODAS, i.e. OWL DL, this semantics cannot be precisely axiomatized. Thus, `Content` about `Entity` and `Content` about `Relation` help to distinguish content units focussing on one single entity from those that describe "something" involving several entities. So, when the user reads a resource that deals with many entities including u , processing a rule containing the adaptation part as shown in 7 leads to recommendations encompassing all those resources that have u as the one and only subject.

Another type of adaptation, which exploits the content semantics as captured by sub-concepts of `Content`, is to go from resources containing pure text to related resources with figures, images etc. that can serve as illustration, i.e. from `Unliteral` `Content` to related `Literal` `Content`. While reading, it may also be helpful to browse through other resources that deal with the current one, i.e. from `Content` to `Meta-Content`. Examples of the type `Meta-Content`, which is axiomatized as `Content` that has `Content` as subject, are the summary or the reference section of an

article. Besides, when the user reads a `General Content`, which deals with a concept, the system may also make recommendations for `Specific Content` which deals with an instance of this concept. Thus, recommendations of this type can be seen as examples that support the comprehension of resources with a rather abstract content.

Content-based Adaptation – Narratively-related contents While the adaption rules presented in the last section exploit the connections implied by a semantic description of the resources' content, this section describes conditions that make use of the given narrative structure. In fact, the order of atomic parts contained by a `Composite Content` resource could reflect a particular relationship (dependency, causality etc.) between the described entities - which might be only in the mind of the author and not directly encoded via properties of the ontology. Using properties modelling the structure of content resources such as `has part`, `is pre` and `is post`, the narrative sequence given by the author can be reproduced. The following conditions lead to resources annotated as the subsequent content of the one currently read:

$$\leftarrow [\dots] \text{AtomicContent}(x), \text{AtomicContent}(y), \text{isPostContent}(y, x) [\dots] \quad (7)$$

Note that the specific subject of the content is not decisive. Here, all resources y that have been placed by the author after x are considered by the system as relevant to the current context.

Task-based Adaptation Similarly to adaptation that is based on the structure of the content, the type of adaptation discussed in the following makes use of the sequence of processes given by the supported workflow. That is, we assume that the tasks supported by the application are specified as a `Computer-aided Process` as discussed in Section 2.3. Based on this notion, the adaptation part can be instantiated in a way that leads to recommendations suitable for the current task. This could be accomplished as follows:

$$\begin{aligned} \leftarrow [\dots] \text{ComputerAidedProcess}(p), \text{ApplicationInteraction}(p1), & \quad (8) \\ \text{isPartOf}(p1, p), \text{User}(u), \text{isInvolvedIn}(u, p1), & \\ \text{ApplicationInteraction}(p2), \text{isPostProcess}(p2, p1), \text{Service}(s), & \\ \text{instrument}(p2, s), \text{ExecutableContent}(y), \text{isRepresentationOf}(y, s) [\dots] & \end{aligned}$$

So, when the `User` is involved in an interaction that is part of a `Computer-aided Process`, then the system recommends subsequent processes as given by the `is post process` property until the user accomplishes the task, i.e. until he/she obtains the output of the corresponding `Computer-aided Process`. More precisely, the system would recommend an `Executable Content`, which is a representation of the respective `Service`. This `Service` acts as the instrument of the subsequent process. Note that for this purpose, the modelling service of the portal tries to identify the current process, i.e. an instance of `Application`

Interaction, or an instance of a sub-concept of `Application Interaction` such as `Executing`, `Reading` etc. Also, the corresponding `Computer-Aided Process` has to be computed. This is a non-trivial task if the system supports many workflows because there are certain types of `Atomic Application Interaction`, e.g. `Browsing`, `Timeout` and `Reading`, that are part of almost all workflows.

Currently, the service attempts to match the type of the current interaction with the type of all the atomic parts of the supported workflows. For instance, there is a `Learning Description Logic-workflow`, which contains a sequence of `Reading` activities that involve articles dealing with description logic formalisms such as `ALC`, `SHIQ`, `SHOIN(D)` and `OWL`. As there are not many workflows currently available in the prototypical implementation of the system, there are no ambiguities. In the future, we aim to reduce ambiguities by considering further information in the matching such as the entity types involved in the respective `Application Interaction`.

The style of adaptation elaborated here exploits only structural information of `Process`. It might also make sense to incorporate more information such as other properties and entity types involved in `Process`. The recommendations would then encompass semantically-related services, and could be achieved in a way similar to what has been done for semantically-related content.

Applying Constraints to the Adapted Set When conditions of the adaptation part are satisfied, the system could generate a large number of recommendations. However, other parts of the adaptation model reduce them to a set of relevant resources. Relevance here is assessed with respect to user and environment conditions. That is, resources are relevant if they meet users' requirements and can be appropriately presented to the user, given the environmental conditions.

Again, conditions that can achieve this result can be specified using ODAS. For instance, a `Content` can be seen as appropriate for a user in case he has `credential` for it, does not know it, has not read it and has `interest` for it (i.e. for entities that this resource deals with). The instantiation of the user-constraint part of the rule may be as follows:

$$\leftarrow [\dots] \text{Content}(y), \text{Credential}(c), \text{User}(u), \quad (9)$$

$$\text{requiresCredential}(y, c), \text{hasCredential}(u, c)[\dots]$$

Note that as negation is not supported in DL-safe rules, negative constraints such as `not know` and `not has read` have to be formulated in a separate rule that results in instances of the form `needsFalsified(user, content)`. Correspondingly, the recommendation service neglects recommendations that have been "falsified".

The last part of rule 2 shows an instantiation of the environment-constraint part. These conditions make sure that the recommended resources are appropriate for being displayed within the user's environment. For example, in case the interaction with the user is taking place via `Voice`, the CBOs suggested to the user must also be an instance of `Auditive`. When the `resolution` and `size` of the client `Display Device` is taken into consideration, only a preview version of the resources may be

recommended. Other user characteristics as well as environmental constraints may be applied similarly.

4 Implementation and Discussion

As a proof of concept, we have implemented a prototype personalized portal using the above described rule-based adaption model. We have already presented this prototype implementation in section 3.1. Preliminary evaluations have shown that recommendations reasonably match the resources that would have been manually chosen by the user. However, evaluating ontology-based applications is far from straightforward. In fact, the widely adopted precision/recall measures only measure part of the relevant aspects of a system.

A relevant aspect which is neglected by precision/recall evaluations is the effort spent in providing the necessary knowledge to the system. In our approach, knowledge has been created manually, resulting in domain ontologies and annotations of resources based on them with high quality. This leads to high quality recommendations but implies a high investment of time and effort to get the system run. In fact, we think our approach (and any ontology-based application) can achieve high effectiveness, but it will be necessary to examine more closely the trade-off between modelling effort spent and effectiveness achieved. In fact, evaluations need always to be sensitive to the amount of background knowledge previously provided to the system. In order to lower the effort, we will in the future experiment with different systems for automatic knowledge acquisition [4] and annotation [8].

A further aspect is related to the fact whether the user is satisfied with the current adaption model of the system. In the current implementation, we experiment with many different adaptation styles, i.e. different configurations of rules. However, as only limited information can be acquired from the user and the environment, the constraints on relevant resources are rather generic and therefore lead to some overgeneration, and thus to a high number of recommended resources. Of these, possibly not all will be really relevant, such that precision will be actually negatively affected. In fact, in the implemented prototype, most of the resources are recommended due to the fact that their content is related to the resources currently inspected by the user. However, only a few of these recommendations have been assessed by the user as really relevant. That is, a high recall has been achieved at the expense of precision. We have tweaked this recall-precision tradeoff by using a subset of the discussed adaptation rules. However, a more principled solution is definitely required here. We think that a sophisticated ranking mechanism can improve precision. We aim to improve this by a prioritization of rules and in particular a mechanism to process weights attached to conditions of the rules in order to compute the "relevance" of the recommendations. Concerning the subjective "correctness" of the rules, we will work on a presentation mechanism that facilitates the comprehension of rules. This will make it easier for the user to adapt them to what he/she deems as correct. Since after all, relevance is subjective, we think this is a way to improve the effectiveness of the system.

Finally, an important aspect is runtime performance. Currently, the inference engine requires several minutes to process and update recommendations, despite using one of the fastest reasoners available (see evaluation of KAON2 in [17]) and a decidable rule

formalism (see discussion on DL-safe rules [18]). In fact, the low performance is also due to the many richly axiomatized concepts used in the rules. In fact, state-of-the-art OWL reasoners including KAON2 do not perform well with the combined use of disjunctions and equality as they open up drastically the space of possible solutions that have to be explored for making inferences. However, eliminating these primitives would result in a loss of semantics and, hence, in recommendations of lower effectiveness and comprehensiveness (recommendations based on less sophisticated models). Thus, the tradeoff between the degree of comprehensiveness of personalization and performance needs to be examined. Improvement can be achieved here by finding the appropriate degree of axiomatization.

Therefore, we will elaborate on these ideas to accomplish improvements. Also, we will work on more comprehensive evaluation of useability that besides precision and recall also considers factors such intuitiveness (controllability), performance as well as the resources (time and effort) needed to run the system.

5 Related work

All personalization models discussed in Section 2.1 are definitely related to the approach presented in this paper. However, we have already argued in Section 2.2 that an approach which builds on ontologies and rules can address several problems common to previous work. Therefore, this section considers only ontology-based approaches.

In this respect, our approach is distinct with respect to the degree of comprehensiveness and generality. Other approaches (see [6], [5], [7], [10], [21]) are mostly geared towards specific problems of adaptation. Frasinca et al. [7], for instance, also employ a domain ontology to model system resources, i.e. ontology individuals for slices and properties for links between them. In their approach, the domain ontology has a different character as it specifically models the museum domain whereas our ontology deals with the domain of adaptive system in general.

In some cases, an explicit conceptualization of resources, e.g. via the concept `Document` is also employed [6][5][10]. However, the comprehensiveness of ODAS is more appropriate for a more fine-grained adaptation. It supports the representation of many models we deem as valuable for sophisticated adaptation (see discussion in Section 2.1). For instance, via `Content` and `CBO`, there is a clear distinction between content-, structure- (`Content`) and presentation-related (`CBO`) aspects of the resource. Therefore, ontology-based adaptive presentation is possible.

In many approaches, the employed ontology is rather a taxonomy of keywords, e.g. the ACM topic hierarchy⁴ [5]. In our approach, resources are represented as entities that describe individuals of a domain ontology. While keywords have no formal meaning per se, the semantics of entities provided by the domain ontology can be exploited by a reasoner to classify resources as subtypes of `Content`, which can be used to recommend semantically-related resources.

Most importantly, the adaptation model introduced here differs substantially from the approaches discussed so far. Adaptation is currently supported by querying the ontologies with corresponding adaptation rules. The rules are essentially conjunctive

⁴ <http://www.acm.org/class/>

queries [12]. However, the formal model we propose is general, domain- and formalism-independent. Personalization in any domains can be achieved by adapting the concrete grounding based on DL-safe rules. Also, we have shown different styles of adaptation that can be accomplished by different variations of this grounding.

6 Conclusion

We have proposed an adaptation model to be used in an ontology-based approach to personalization. This model has been formalized in an abstract, formalism-independent way. Also, a concrete grounding based on DL-safe rules has been provided. A comprehensive ontology capable of capturing many aspects relevant to adaptation has been exploited to express conditions in the DL-safe grounding. Many variations of this grounding, i.e. adaptation rules, have been discussed to illustrate the use of the model. We have shown that many different styles of adaptation can be achieved this way. In particular, the concept of adaptive content, adaptive structure and adaptive presentation can be supported by means of a declarative model of the adaptation rationales. The rules that make up this model can be modified (also by the user) for specific adaptation requirements. Also, we have argued that this approach can solve many drawbacks in current systems. We have further also discussed the bottlenecks and weaknesses of our approach.

References

1. M. Balabanovic and Y. Shoham. Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
2. P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.
3. P. Brusilovsky and D. Cooper. Domain, task, and user models for an adaptive hypermedia performance support system. pages 23–30. ACM Press, 2002.
4. P. Cimiano and J. Völker. Text2onto - a framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 227–238, 2005.
5. P. Dolog, R. Gavriloaie, W. Nejdl, and J. Brase. Integrating adaptive hypermedia techniques and open RDF-based environments. In *Proceedings of 12th International World Wide Web Conference*, 2003.
6. P. Dolog, N. Henze, W. Nejdl, and M. Sintek. The Personal Reader: Personalizing and Enriching Learning Resources using Semantic Web Technologies. In *Proceedings of the Adaptive Hypermedia 2004*, pages 85–94. Springer, 2004.
7. F. Frasincar, G. Houben, and R. Vdovjak. Specification framework for engineering adaptive web applications. In *Proceedings of the 11th International World Wide Web (WWW) Conference*, 2002.
8. S. Handschuh and S. Staab, editors. *Annotation in the Semantic Web*. IOS Press, 2003.
9. N. Henze and W. Nejdl. Adaptation in open corpus hypermedia. *Special Issue on Adaptive and Intelligent Web-Based Systems*, 12:325–350, 2001.
10. N. Henze and W. Nejdl. Knowledge modeling for open adaptive hypermedia. In P. De Bra, P. Brusilovsky, and R. Conejo, editors, *AH*, volume 2347 of *Lecture Notes in Computer Science*, pages 174–183. Springer, 2002.

11. J. Herlocker, J. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the ACM conference on Computer supported cooperative work*, pages 241–250. ACM Press, 2000.
12. I. Horrocks and S. Tessaris. Querying the semantic web: a formal approach. In *Proceedings of the First International Semantic Web Conference (ISWC)*, volume 2342, pages 177–191. Springer, 2002.
13. G. Houben, P. Barna, F. Frasincar, and R. Vdovjak. Hera: Development of Semantic Web Information Systems. *Proceedings of the Third International Conference on Web Engineering*, pages 529–538, 2003.
14. U. Hustadt, B. Motik, and U. Sattler. Reducing shiq description logic to disjunctive datalog programs. In *Proc. of the 9th International Conference on Knowledge Representation and Reasoning (KR)*, pages 152–162, 2004.
15. J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
16. K. Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pages 331–339. Morgan Kaufmann, 1995.
17. B. Motik and U. Sattler. A comparison of reasoning techniques for querying large description logic ABoxes. In M. Hermann and A. Voronkov, editors, *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 4246 of *Lecture Notes in Computer Science*, pages 227–241. Springer, 2006.
18. B. Motik, U. Sattler, and R. Studer. Query Answering for OWL-DL with Rules. In *Proceedings of the International Semantic Web Conference (ISWC)*, pages 549–563. Springer, 2004.
19. J. Nielsen and U. Lyngbaek. Two field studies of hypermedia usability. *Hypertext: State of the Art*, pages 64–72, 1989.
20. L. Terveen and W. Hill. Beyond Recommender Systems: Helping People Help Each Other. *HCI In The New Millenium*, Carroll, J. ed. Addison-Wesley, 2001.
21. T. Tran, P. Cimiano, and A. Ankolekar. Rules for an ontology-based approach to adaptation. In *Proceedings of the 1st International Workshop on Semantic Media Adaptation and Personalization*, 2006.
22. T. Tran, H. Lewen, and P. Haase. Semantic software engineering: the role and usage of ontologies in information systems. *Proceedings of the 5th IEEE International Conference on Computer Science - Research, Innovation and Vision for the Future*, 2007.
23. J. van Ossenbruggen, J. Geurts, F. Cornelissen, L. Rutledge, and L. Hardman. Towards second and third generation web-based multimedia. In *Proceedings of the 10th International World Wide Web (WWW) Conference*, pages 479–488.