Novel Approaches to Acquisition and Maintenance of User Model

Anton Andrejko*
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
andrejko@fiit.stuba.sk

Abstract

There are many problems related to the Web, e.g. one is overloaded by huge amount of information or a searcher can get lost in information space. These problems can be partially overcome by employing personalization. We focus on the user modeling area and we exploit existing knowledge, mostly from the Adaptive and Semantic Web. We focus on creation and maintenance of the user model. We propose three novel methods for acquisition and maintenance of user characteristics in the user model. The first method is based on *generating questions* to be used for user model.The second method is based on the *content analysis* and assumes that comparing attributes of documents, which were found interesting for a user, can be a source for discovering information about user's interests. The third method is based on *spreading activation*. If there are connections between information concepts of the domain model user's characteristics can be utilized even for concepts that have not been visited yet. The proposed methods were evaluated by means of software tools that were incorporated in research projects.

Categories and Subject Descriptors

D.2.13 [**Reusable Software**]: Reuse models; H.3.3 [**Information Search and Retrieval**]: Selection process; H.3.4 [**Systems and Software**]: User profiles and alert services; H.5.4 [**Hypertext/Hypermedia**]: User issues; K.3.1 [**Computer Use in Education**]: Computer-assisted instruction

Keywords

adaptation, concept, instance, personalization, ontology, semantics, similarity, spreading activation, user modeling, user characteristic

*Recommended by thesis supervisor: Prof. Mária Bieliková. Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on February 19, 2009.

1. Introduction

Information technologies have become a part of our lives in recent history and using services provided by the Web have become especially popular world wide. One limitation is that not everybody is computer-literate and able to find the appropriate information effectively. Even though the Semantic Web introduces solutions to some problems, there are still many problems that limit the speed and extent of its expansion. Problems can be partially overcome if we take into account a user's particular search characteristics and we exploit existing search optimization approaches that are currently used in adaptive web-based applications. The user model in adaptive web-based applications consists of identifying user's characteristics that are used for personalization of *layout*, *navigation* or *content*. There are several approaches used to acquire user characteristics and keep them up to date. One method is to ask the user *explicitly* or observe the user's behavior while working with the application (*implicit feedback*). Another useful approach is to *mine* the user characteristics from logs, which can be processed on client-side and/or server-side.

We present a contribution to the current state of the art in the user modeling area, namely we focus on creation and maintenance of the user model. We propose three novel methods to acquisition and maintenance of the user characteristics in the user model. The first method is based on *generating questions* to be used for user model updates. Particular questions are generated according to the analyzed properties of the information concepts that are the subject of the observed application domain. The entire process of asking questions is driven by user-defined rules. The second method based on the *content analysis* assumes that comparing properties of documents, which were found interesting for a user, can be a good source for discovering information about user's interests. Moreover, in this methodology we impute reasons that might have caused user's interest in the content. The last method is based on *spreading activation*. If there are connections between information concepts (e.g., learning objects in an educational application) of the domain model user's characteristics can be utilized to extend and extrapolate beyond the known characteristics, even for concepts that have not been visited yet.

The methods were evaluated by software tools that were incorporated in research projects aimed at job offers (project NAZOU [24], `http://nazou.fiit.stuba.sk/`), digital libraries (project MAPEKUS [6], `http://mapekus.`

`fiit.stuba.sk/`) and learning programming domain (project PeWePro [7], `http://pewepro.fiit.stuba.sk/`) that have been conducted successfully at the Faculty of Informatics and Information Technologies at Slovak University of Technology in Bratislava in the period of 2004–2008.

2. Thesis Objectives

We propose novel approaches to automatic acquisition and maintenance of the user characteristics that employ semantics provided by ontological representation. The main objectives of this work are following:

- We assume that questions can be generated automatically according to the concepts in the domain model to be used for acquiring and maintenance of user characteristics for the user model.
- We assume that comparing properties of documents, which users found interesting, leads to discovery of information about users' interests. Our goal is to propose a method that uses content analysis (similarity estimation) to determine information suitable for the user model.
- We assume that, if well-defined connections between information concepts are available, a user's characteristics can be changed for the concepts, even if they have not been visited yet by the user. Our goal is to propose a method for maintenance of user characteristics in the user model based on relationships among concepts in the domain model.
- Experimental evaluation of the proposed methods with regard to their domain independency.

3. Adaptive Semantic Web

A way to improve efficiency in information acquisition is a personalized approach based on user's particularities aimed at adaptation of the content, layout or navigation in the information sources. The adaptation in an adaptive application is a modification of content, style of presentation and navigation, whereas the personalization is an adaptation that is provided to the particular user.

Presently, most of the available information in the Web is provided in a form primarily suitable for human beings, where the choice of its representation and presentation is up to individual information providers. Therefore, the next stage in improving efficiency of information processing is adding semantics to the content (e.g., the Semantic Web).

3.1 Models of Adaptive Applications

There are two terms used very often – *adaptivity* and *adaptability*. A common feature for both terms is cooperation with the user. However, in an adaptive information source a user's behavior needs to be observed to obtain necessary information for personalization. In an adaptable information source, the user's action is required, e.g. setting up properties. Adaptive applications consist of four main parts:

- *domain model*, which represents the scope for which the application is designed to be performed;
- *user model*, where all the actual user's characteristics, which are necessary for adaptation, are stored;
- *adaptation model*, which specifies a method for the adaptation to be accomplished;
- *navigation model* describes user's possible moves among the concepts within the information space.

Since the purpose of the adaptive applications is to provide personalization, the user model is the most important part. There are more definitions of the user model which differ according to the way the model is used. The user model represents beliefs about the user that include preferences, knowledge and attributes for a particular domain [20]. We use the common term *user characteristics* to describe preferences, knowledge, attributes and other identifying features that are included in the user model. The user's environment together with the user model is called the *context model*.

Adaptation in the web-based applications occurs on three levels, namely adaptation of the *content*, *presentation* [27] or *navigation*. In Brusilovsky's works [8, 9] we find techniques for adaptation divided in two groups. The first group aggregates techniques related to the adaptation of the *presentation* (including content). The second group contains techniques for adaptation of the *navigation* (see Figure 1).

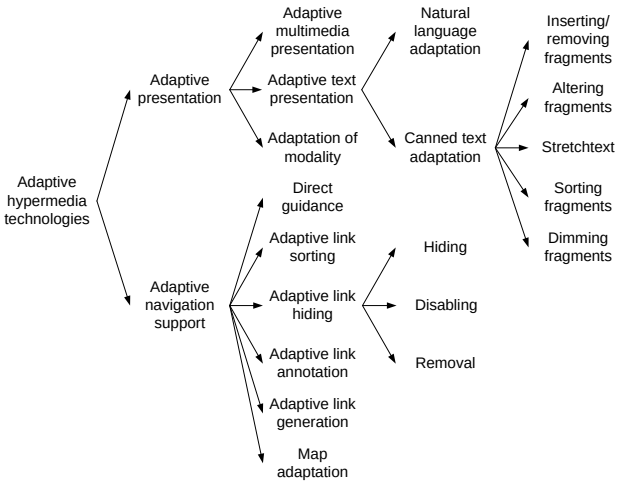

Figure 1: Taxonomy of adaptive hypermedia technologies [9].

A detailed description of these techniques is beyond the scope of this work and is provided in [8, 9]. For the content adaptation Brusilovsky recognizes and utilizes methods like *conditional content*, *alternative content*, or *sorting of the content*. It is possible to realize these methods by using various techniques, namely *inserting*, *removing*, *altering*, *sorting*, *dimming fragments* and *stretchtext*.

In this work we focus on adaptation based on the user model (i.e., personalization) so it is important to note that the user model must be also adapted to be continually up to date and to provide proper information about the user for further personalization.

3.2 Adding Semantics to Adaptive Applications

When a content presented to the user is personalized, it is still suitable only for a human because it is designed for use by humans. One solution is to represent the Web content in a form that is easily *machine-understandable*. Here

emerges Tim Berners-Lee's vision known as the Semantic Web initiative [5] that tries to add semantics to knowledge to make it processable by automated tools as well as by people. Probably the easiest way to capture the meaning of the content is provided by metadata. However, authors of the content can use own terms to describe semantics of the presented information, with resulting variation in metadata descriptive elements which causes problems for heterogeneous applications using various information sources.

A more complex solution than metadata provides an ontology. The term *ontology* originates from philosophy, namely from the study of the nature of existence. We use Studer's et al. definition [26], which extends Gruber's definition [16] where an *ontology* is a formal, explicit specification of a shared conceptualization. In the ontology we define *classes* (e.g., general things), *instances* (particular things), *relationships* among those things, *properties* (and property values) of things, *functions*, *constrains* and *rules*. This gives the ontology much more powerful expressiveness than the metadata approach has. Another term, which is used very often along with ontologies, is a *concept*.

An example of an instance representing a part of job offer in a job offer application domain is depicted in Figure 2. Rectangles used in Figure 2 represent instances of concepts. Every such an instance has its unique identifier, but we present only its label for clarity (e.g., *Salary*). *Datatype* and *object properties* are used to assert specific facts about instances. Datatype properties express relations between concept instances and RDF literals and XML Schema datatypes. Object properties express relations between two instances. To distinguish between object and datatype properties a dashed line is used for datatype properties (e.g., *maxAmount*). *JobOffer* is the instance identifier which several properties are connected to. For simplicity, we present only a few of them, and surround multiple properties (e.g., property *hasPrerequisity*) by a rounded box.

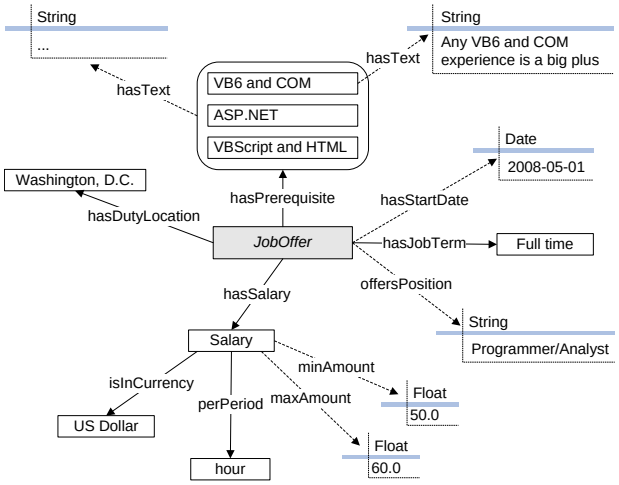

Figure 2: Example of an instance representing a part of job offer.

In adaptive web-based applications, the ontologies can be used as a mean to represent models. We focus particularly on the *user model* and also on *domain model* which it is interconnected with.

4. User Modeling

A user model represents various user characteristics that can be used to adapt the *content*, *presentation* or *navigation* of the informational materials presented. A content of the user model is described with variety of terms, namely *attributes*, *features*, *characteristics* or *properties* are used frequently. We use the term *characteristic*. The more relevant characteristics describing the user that are included in the user model, the more accurate and useful is the personalization provided by the adaptive application.

There are several approaches that can be employed in the user modeling process. We discuss approaches (e.g., stereotype model, overlay model) which are the most commonly used and look at user modeling from different perspectives.

Stereotype model. The stereotype is a collection of frequently occurring characteristics of users [25]. Users are associated with one or more stereotypes that commonly reflect a user's knowledge, social background, computer experience etc. The stereotype model is prone to inaccuracy due to the need for heavy reliance on inferences. Even for a stereotype that has been selected and applied correctly, some inappropriate characteristics can be inferred for particular user. Here, it is important to emphasize that in this case the personalization is not performed to the benefit of particular user, but to the selected stereotype in a collective sense.

Overlay model. The main idea is that a user's knowledge of the subject is relevant subset of the domain knowledge and the overlay user model is an overlay over the domain model [19, 18]. Any domain concept from domain model has a corresponding value in the overlay model which represents the user's knowledge of the concept. The main drawback is the necessity of initialization. The way to overcome this problem is a combination of overlay and stereotype user model. There are also inherent drawbacks, e.g. each application uses its own (different) representations, terminology, granularity of stored information, requirements for higher or lower safety, etc.

4.1 User Model Representation

There are several approaches to representing and storing a user model in web-based applications. We do not discuss representations that use proprietary formats as this would prevent the sharing and reuse of the user model.

Non-ontological representations. Using a relational database is quite straightforward, offers good performance, and several other advantages such as security, distribution ease, data recovery, etc. Another frequently used approach is the representation of the user model by an XML based language. Both mentioned approaches offer only a way to describe user characteristics and do not offer any added value from the user modeling perspective. An ontology-based approach to user modeling offers a way to move user modeling from the low-level description of user characteristics to a higher level with additional possibilities.

Representing user model by ontology. The advantages leading to using ontologies for user modeling come from the fundamentals of this formalism. Ontologies provide a common understanding of the domain to facilitate reuse

and harmonization of different terminologies [20]. These characteristics support reasoning, which is considered as an important contribution to the ontology-based models. By creating an ontology-based user model we increase the probability that user characteristics will be shared among a range of systems of the same domain.

4.2 Acquisition and Maintenance of User Characteristics

We have already mentioned that relevant and up to date characteristics describing the real user are needed in the user model to perform accurate personalization. Processing of acquisition and maintenance of characteristics roughly follows these steps:

1. *Sources identification.* At this stage we need to identify sources from which we can extract information for the user model.

2. *Collecting information.* There are two basic approaches to collecting information about the user – *explicit* and *implicit feedback* [11].

3. *Analyzing and extracting information.* Information acquired by observation of the user's behavior or implicit feedback requires further processing before it is stored in the user model as a characteristic [4].

4. *Changing the user model.* User characteristics are stored in the user model. This process is known as *initialization* if there is no pre-existing characteristic in the user model; or *maintenance* in the case when that characteristic already exists in the user model and only its value is changed.

5. Acquisition of Characteristics Based on Questions

The method employs direct responses from the user. It can be accomplished by filling in forms, answering questionnaires, questions, etc. We distinguish questions that can be answered by filling their missing part or can be answered with a positive or negative answer. We deduce that a characteristic of the user, we are interested in, is a part of the question. Each question contains the *name of a characteristic* from the user model. The answer can be a number, text, ordinal value, etc. The name of the characteristic can be also contained in the answer.

At the very beginning, a list of properties for a concept from the domain model is acquired. Each property relates to a characteristic for which a question will be generated (i.e., it will appear as an instance of a characteristic in the overlay user model). Afterwards, properties can be used for *acquisition*, if an instance of respective characteristic does not exist yet in the user model, or *maintenance*, if an instance of respective characteristic already exists. To be able to generate questions we employ templates. Instead of the name of the characteristic a special identifier is used in the template. Following context relates to the question – *circumstances*, which invoke generating of the question and an object specifying *which* question should be generated and *what about*. The circumstances of the question are handled by user defined rules and the object depends on the particular characteristic being researched for the user model.

Unique names and priorities assigned to a question along with its structure are stored in *Characteristic binding*. Furthermore, a binding specifies a *template* and a *noun* which will be used in the generated question as a name of the user characteristic. A suitable template for the characteristic is selected from *Question templates* according to its unique identifier. Words to be used in the question are stored in *Vocabulary*. The user is also provided with a part of an answer in the case the answer is of an *ordinal value* or an *option* type. If we deal with the maintenance, properties are acquired with regard to *rules fulfillment*. Employing a priority hierarchy also allows response to the cases when it is appropriate to suppress the generation of a question.

5.1 Binding Characteristics

Binding characteristics provides additional information for the user model that is necessary to generate a question. Such additional information for a characteristic are unique identifier, priority, template and noun used in the sentence, which specifies a name of the characteristic. A characteristic can be represented as a set of concepts. When binding its structure we consider namespaces, classes, datatype or object properties, and instances. Simple characteristics are represented as RDF triples. Such a structure is easy to bind, but this is not applicable to more complex user characteristics where characteristics can have relationships to others, can create taxonomies, or a characteristic which can get an ordinal value; or which can be represented as a class or an instance.

We propose a way of assigning names to instances in regard to the structure of concepts. The instances can be named by *using pattern* – an instance is created according to the defined pattern with a timestamp suffix, *adjusting desired value* – an ordinal, text or numerical value can be assigned, and *name use* – a defined name is used to name an instance.

We use unique names (i.e., keys) to represent particular bindings which are stored in *Characteristic binding*. A name of a key consists of an *identifier* specifying a type of binding and a *postfix*, which is a unique number assigned to the key. There are several possible identifiers that can be used, namely *c* (class), *op* (object property), *dp* (datatype property), or *i* (instance).

In the Figure 3 an example is depicted with keys assigned to individual parts of characteristic's structure.

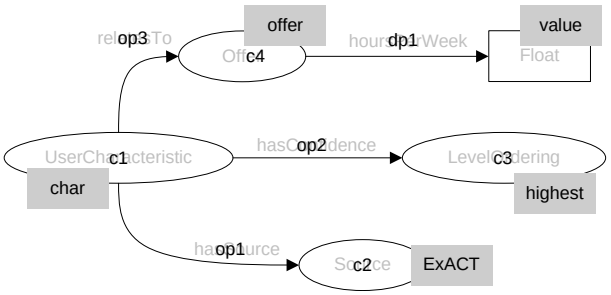

Figure 3: A characteristic with assigned keys.

5.2 Rules for Question Generation

An advantage of this approach is simplicity of creating and maintenance of such rules. We employ a well known

principle for rules: *IF* [condition] *THEN* [action]. Required parts of the condition are a characteristic from the user model, operator, and parameter of the condition. Default action generates a question. A condition is defined according to data types, which represent a value of a characteristic. Literal values can be of several types defined according XML Schema. An instance can get an *ordinal value* or an *option*. Operators, such as *less*, *greater*, *equal* as well as special operator *instanceOf*, can be used in a condition. Well defined rules initiate generation of a question in a right time, e.g. if a characteristic is older than given time period. Another example is generating questions with regard to a tool that is responsible for creating or updates of the characteristic. Employing priority allows also reaction to the cases when it is appropriate to suppress the generation of a question.

5.3 Discussion

The proposed method is based on generating questions in natural language (here English). We employ concepts from the domain model in the process of question generation. To evaluate the proposed method a software tool called Explicit Actualizer (abbreviated *ExACT*) was developed. The method was evaluated using the job offer domain ontology (created in the course of the project NAZOU) and ontology of scientific publications (project MAPEKUS) with respecting user models (represented as ontology). In both domains, we bound selected characteristics and created templates to generate questions. Furthermore, rules to manage generating questions were defined. The templates were proposed with regard to their universality, i.e. our goal was to design templates to be used for as many questions as possible. However, the case when too many templates are provided is not very reasonable, e.g. creating one template for each question is not very effective. We designed 7 templates for the job offer domain that allows for generation of questions addressing 22 characteristics. For the scientific publications domain we proposed 3 templates which allow generation of 5 questions only what is influenced by the straightforward structure of the domain.

The main problem of feedback based solutions is that necessary questions are mostly hard coded for a particular application. However, there are some solutions that use generation of the questions, such as dialog systems. Dialogue systems can provide relatively open-ended prompts, i.e. users may provide responses that are not an exact fit with what is expected [23]. Employment of ontology helps to exploit well known relationships to achieve more natural communication. Sentences are generated according to templates. The type of the template as well as the use of proper terms is driven by rules. In [22] there is described an approach to planning utterances in natural language that can dynamically use context information about the environment in which a dialog is located. The educational system OntoAIMS [13] is built upon the OWL-OLM. OWL-OML uses a *dialog agent* to maintain dialog with regard to the prepared questions. The questions are generated according to the domain concepts and can be answered by transforming them into queries. Our method generates questions according to a concept's properties and provides the user with prepared answer considering the type of the question. The M-PIRO project uses meta-data to generate text in natural language [3]. Unlike this project, our method generates questions with regard to concepts.

6. Acquisition Based on Content Analysis

We present a method for the comparison of instances of ontological concepts aimed at the identification of common and different aspects for personalization purposes. The method exploits the advantage of ontological information representation and computes instance similarity with regard to particular properties of concepts. In personalized applications where the user model is available, the method also supports more accurate similarity computation for individual users according to their characteristics. Our method is designed with regard to domain independency to be available for using in arbitrary domains.

6.1 Recursive Traversing of Ontology Instances

The main idea of the method comparing ontology instances is based on the evaluation of common property pairs present in both instances. The rough principle of the method illustrating comparison of two instances *instanceA* and *instanceB* is shown in Algorithm 1. Detailed description of the method is provided in [2].

Algorithm 1 Recursive method basics

```
function getSimilarity(instanceA, instanceB)
    set similarity to 0.0
    set counter to 0
    store properties for instanceA and instanceB
    to properties

    foreach property in properties do
        increment counter
        if property is in both instances then
            store connected elements
            to elementX and elementY
            add
            computeSimilarity(elementX, elementY)
            to similarity
        else
            add 0.0 to similarity
        end if
    end foreach

    return similarity/counter
end function

function computeSimilarity(elementX, elementY)
    if property is datatype then
        return
        getDatatypeSimilarity(elementX, elementY)
    else
        set similarity to 0.0
        add getObjectSimilarity(elementX, elementY)
        to similarity
        add getSimilarity(elementX, elementY)
        to similarity
        return mean value of similarity
    end if
end function
```

When comparing two instances of the concepts, properties can appear as single in both instances, multiple in both instances, or single/multiple in one instance only.

When the property has a single occurrence in both instances, then the similarity of related elements (*instances* in the case of object properties or *literals* in the case of

datatype properties) is evaluated using different similarity metrics. The comparison of *datatype* properties ends after a metric is used to compute the similarity measure between the related literals. For *object* properties a metric for related instances is used (e.g., *taxonomy distance*) and further comparison is performed recursively on the respective instances until literals are reached or until there are no properties left.

Multiple occurrences of properties (e.g., *hasPrerequisite*) in an instance are the most complex case we address. In this case, two sets are constructed which contain elements connected to the examined property in the first and second instance respectively. These two sets can have different cardinalities – the problem is to identify similar elements between these two sets. We use our similarity measure to identify such element pairs, which are then compared and the computed similarity contributes to the total similarity between the two instances.

However, the identified pairs do not provide satisfactory results in some cases. For example, if in the first instance the *hasPrerequisite* property has the value *"Java or C programming"* and in the second instance multiple values *"Java programming"* and *"C programming"* consistent results are difficult to achieve. In our approach a pair with higher similarity according to the used similarity metric is selected (i.e., similarity with only one property's value from the second instances is considered), but more complex heuristics can be proposed and employed to identify a $1 : n$ mapping.

If single or multiple occurrence of a property occurs only in one instance, we estimate similarity of values attached to the property as equal zero. It is based on the similarity definition, i.e. the similarity equals zero if two objects are entirely different. Here, we assume that instances are entirely different in the property, since a value is assigned to the property in one instance only.

A variety of comparison metrics can be used to compute similarities between instances or literals connected to a property. We proposed two groups of metrics with regard to a property's type since they must be treated differently due to their different nature – *datatype* and *object* metrics. The description of the metrics is beyond the scope of this paper.

6.2 Similarity Estimation

The total similarity between two instances of ontological concepts is aggregated as the mean value of the similarities computed between elements connected to particular properties. The computed similarity is symmetric. Let *PropertySM* be a similarity measure (SM) that is computed for elements connected to a common property. Then similarity measure $sim\,(InstA, InstB)$ for two instances is computed as follows:

$$\frac{\sum_{i=0}^{|A\cap B|} PropertySM_i\,(elementA, elementB)}{|A \cup B|}, \qquad (1)$$

where *elementA* and *elementB* are elements (instances or literals) connected to the i-th property. Since there can be datatype or object properties, we introduce the *General similarity measure* that encapsulates all the similarity measures that are available. It is computed for elements connected to a property (e.g., *PropertySM* used in the Equation 1 is its special case). The *General similarity measure* is computed with regard to the property type. In the case of a datatype property the used metric depends on the corresponding literal type as described above. For object properties, the similarity measure for related instances is computed as the aggregation of the *label-based*, *property-based* and *taxonomy distance similarity measure*.

6.3 Personalized Similarity and User Characteristics

The aggregate of partial similarities is always the same no matter what the context is. To improve the accuracy of our similarity evaluation method with respect to individual users' preferences (if a user model is available), we introduce weights that personalize the similarity estimation which are computed for particular properties and allows us to compute personalized similarity for individual users. The *weight* gets a value in the range $\langle 1, w\rangle$ based on the match between the property and the value of the corresponding characteristic in the user model.

For personalization purposes, our goal is not only to compute the similarity between instances but also to investigate reasons that "caused" the similarity or difference. User preferences can be deduced from implicit and explicit user feedback (e.g., rating). We assume that if the instance includes a property whose value the user likes, it will likely influence his/her rating towards the higher (or positive) values. On the other hand, properties of the content with the values that the user dislikes will influence rating towards lower (or negative) values.

Since we are interested in properties that significantly influence user rating and thus also total similarity, we introduced two threshold values that divide properties into three sets based on the computed similarities. If the similarity computed for a property is greater than the *positive threshold* then the property is assigned to the positive set, if the computed similarity is lower than the *negative threshold* the property is assigned to the negative set.

6.4 Discussion

Experimental evaluation was performed using the software tool called Concept Comparer (abbreviated *ConCom*). The evaluation was performed on the job offer ontology. In the first case, only properties that are common for both instances are considered, thus other properties are ignored and do not influence the total similarity. Using only common properties resulted in a narrow range of similarity values – in 89 % of the cases the similarities were in the range 0.30 to 0.75. We set the positive threshold experimentally to 0.65 and the negative threshold to 0.25, but such thresholds do not produce useful properties that could be used for user characteristics discovery. Therefore, similarity computed for all properties must be used to acquire properties based on thresholds.

In the second experiment, a user who assessed similarity was involved. Similarity computed for common properties was used for comparison with human estimation since its values more accurately mimic human assigned similarity values. For experiments where the user model was involved we used similarity computed only for common properties. The aim of the last experiment was to investigate how the user model influences similarity computation. If the compared properties are similar (i.e., for high

values of the similarity measure) the personalized similarity increases towards higher values (a positive change in the figure), while if they are different, the personalized similarity decreases to lower values (a negative change).

In the following text we compare proposed method to existing approaches. A method computing similarity among instances within an ontology is described in [1]. The data layer estimates similarity by considering simple or more complex types, such as integer and string. We propose datatype similarity metrics that deal with all types of datatype properties as defined by XML Schema. An approach to ontology matching based on instances is described in [21]. Its main idea is to derive similarity between concepts from the number of shared instances, since the number of instances is usually greater than the number of concepts. Concepts are matched with regard to trigram similarity of their names, though experiments showed that it is not very effective due to the high diversity in the concept names. A two phase method for instance comparison of tourism ontology concepts is described in [15]. The advantage of this approach is that total similarity for more than two concepts can be expressed as one number. Furthermore, the similarity of concepts from different contexts can be computed as well. The main drawback of this method is that a similarity ontology holding similarity relations between properties and entity names from the domain ontology must be provided in order for the similarity graph to be built.

We also performed experiments in the scientific publications domain to investigate domain independence of our method, where the computed similarity can be useful, e.g. for clustering algorithms, semantic annotation tools or repository maintenance tools as well as for the recommendation of similar content in recommender systems. The aim here was to improve semantic search using personalized navigation within ontology instances that represent metadata of large information spaces [27].

7. Maintenance Based on Spreading Activation

With regard to defined connections among the concepts in the domain model, we spread a change to other related parts of the domain model, even when the user has not worked with these parts. In this section we describe a method for maintenance of user characteristics that uses spreading activation. Spreading activation principle originates from psychological studies of human memory operations. The principle is based on an idea that initial energy (activation) of a selected node (in a graph) is spread to other nodes.

Our method was primarily proposed for educational domains where relationships among concepts and their fragments are better defined than for the concepts involving a job offer domain. However, the proposed method is not restricted to be used only in the educational domain.

7.1 Models of Adaptive Web-Based Educational System

Existing domain models usually consist of mutually interconnected learning objects that represent learning materials [10]. Having reusability of the models in mind we divide the domain model into a *knowledge item space* and a *learning object space*. An example of the domain model is depicted in Figure 4. The learning object space consists of *learning objects* and *relationships* among them. This part of the domain model meets the standard view of the domain model for adaptive web-based systems. A knowledge item (KI) represents a topic or a key word that represent key terms of the domain. Its aim is a categorization of available learning objects into knowledge items according to learning goals of particular learning objects.

Our approach also supports changing learner's characteristics for the learning objects, even in the case they have not been visited yet, as a knowledge item can be connected to learning objects using relations *prerequisite*, *contain* and *isRelatedTo*.

We use an *overlay* user model that models the relation of the user to the individual parts of domain model. It consists of a *domain independent part* and a *domain dependent part*. The domain-dependent part consists of records about user's visits, interests and knowledge in current educational course.

7.2 Maintenance by Spreading Change

While a user works with a learning object we are obtaining his/her characteristics for that learning object (e.g., estimated interest). Since there is a connection between learning objects and other parts of the domain model we spread a change to other related parts of the domain model even though the user has not worked with these parts yet. Modeling user's characteristics consists of the following steps:

1. Setting respective characteristics for actual learning object (e.g., *Example with I/O operations* in the Figure 4) – with regard to user's activity with the learning object we can find out user's *interest* and *knowledge* for that learning object.
2. Spreading changes of characteristic's values from the actual learning object to knowledge items (i.e. knowledge item *Display* and *Scan*).
3. Spreading changes of characteristic's values from the knowledge item (*Scan* and *Display*) to other related parts of knowledge item space (*Input/Output* and *Files*).
4. Spreading changes of characteristic's values from the knowledge items, which characteristics have been changed, to related learning objects (i.e., see *Example with writing to file* and *Example with displaying items of array*).

7.3 Discussion

For the evaluation process a simple programming course was created. It consisted of 16 learning objects containing simple text, exercise and explanation types. Knowledge items, which learning objects are assigned to, come from ACM classification. The knowledge item space contains 1 476 topics and 4 keywords added specially for our course. We developed an adaptive web-based educational application that recommends learning objects according to defined domain model. The user model is created automatically based on user's activity (i.e., number of learning object's visits, time spent by reading, interest).

We describe the evaluation on spreading interest which is performed on actual learning object with regard to user's

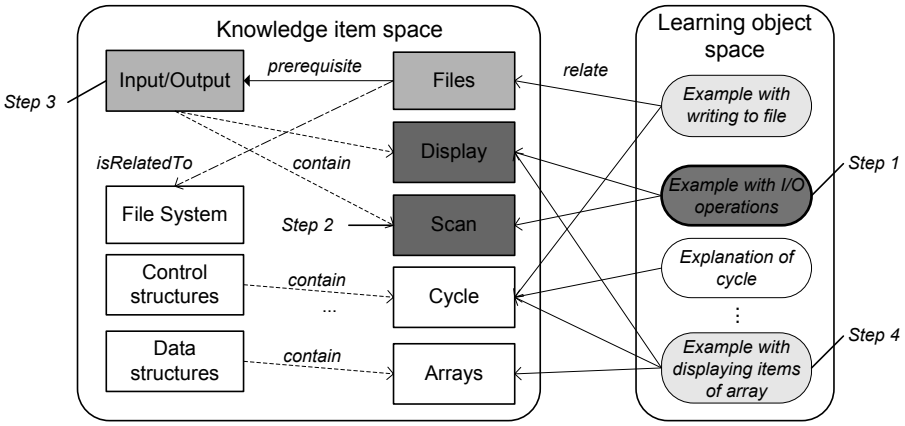

Figure 4: Sequence of steps in process of change of characteristic.

feedback and consequent spreading of changes in the domain model. The interest gets a value from the range −3 (the lowest user's interest) to 3 (the highest interest). The evaluation starts with the empty user model (i.e., with no initialized characteristics). We gradually set user's interest to 3 (to observe changes of interest for actual learning object), then to 2 (to observe spreading change and combination with previous value of interest) and at last to −3 (to verify changes also for negative values). In the first experiment, the interest was adjusted for the actual learning object and afterwards it was spread to respective knowledge items, then continued spreading in the knowledge item space and, at last, spreading of interest continued to respective learning objects for changed knowledge items. In the second experiment the value of interest influenced by its initial value is tested.

There are several ways to model characteristics in a user model. Mostly used are *stereotype* (e.g., Multibook) or *layered* (e.g., AHA!, NetCoach) user models. In Multibook the stereotype can be changed after successfully passed test. The presentation is adjusted to a stereotype not to a user. In the layered user model the evaluation of user characteristics' for a visited document and for related parts of the course (characteristic propagation) is provided. Actualization can be based on author defined rules [12] or on analysis of learner's activity. An alternative approach to rule based adaptation in educational hypermedia systems is described in [17]. Unlike our approach, content to be presented is selected according to the suitability function that estimates the suitability of a learning object for a specific learner. We adjust characteristics and then provide the user/learner with the content with the highest relevance. AHA! provides a propagation of a characteristic's change to other parts of a course via prerequisite relationships between concepts [12]. The propagation is based on author's defined rules. Defining rules for each document (or type of documents) is complicated and time consuming. Furthermore, maintenance of the rules is necessary anytime a new document is added and relations with other documents need to be updated. Personal Reader is an experimental environment supporting personalized learning based on semantic web technologies [14]. The user model (here learner profile schema) provides slots for information about a learner and is represented as a RDF document. Provided recommendations are personalized according to the current learning progress of the user. The entire process of recommending is driven by rules. Our method uses overlay user model and maintenance of user's characteristics is based on spreading activation, i.e. we employed structure of the domain model to spread changes of user characteristics. Our approach allows modeling user characteristics in the entire domain model and not only in the already visited parts, thus, it provides more accurate recommendations of learning objects to achieve more effective education.

8. Contributions

Main contribution of this work is a proposal of three novel methods to automatic acquisition and maintenance of user characteristics that employ semantics provided by ontological representation. Another contribution of this work is its aim at the methods for acquisition and maintenance of user characteristics since, currently, there is less attention paid to these problems (in comparison to adaptive navigation and presentation) in the adaptive hypermedia field.

Method for acquiring user characteristics based on questions. The advantage of the method is that questions are generated automatically for concepts from the domain model and afterwards answers are transformed into characteristics. The entire process (i.e., *when* and *what* question needs to be generated) is driven by user defined rules that can be restricted by question's priority. The method was evaluated using the job offer and scientific publications domain ontology using our software prototype *ExACT*. We designed 7 templates for the job offer domain which allow generating questions for 22 characteristics. For the scientific publications domain we proposed 3 templates which allow generating 5 questions.

Method for acquiring user characteristics based on content analysis. The method is aimed at comparison of instances of ontological concepts. The final similarity is the aggregate result of the individual similarities computed for particular properties while their type is considered to select a suitable similarity metric for each property. The introduction of similarity metrics for properties allows us to take advantage of semantics provided by ontological representation, which allowed us to extend similarity with personalized weights reflecting users' individuality. Furthermore, we investigated reasons (properties) that influenced user evaluation of content (e.g., interest). We have developed the software tool *ConCom* that automates the proposed method. Our experiments showed that similarity where all properties are considered is more suitable

for discovering user characteristics, while similarity computed for common properties only better mimics the similarity estimated by real users.

Method for characteristics' maintenance based on spreading activation. The method is aimed at maintenance of user's characteristics where spreading activation is used. We described our extension of domain model that enables more accurate adaptation using two related parts – knowledge item spaces and learning object spaces – that can be reused across several educational applications. Information about the user is in the user model expressed as knowledge or interest about particular learning objects. We developed a prototype of an adaptive web-based educational application that recommends learning objects from the domain model. Benefit of this approach is in changing user's characteristics in the entire domain model and not only in the parts that the user has already visited. Results of our method are used in the process of personalization and thus more accurate recommendation of learning objects is performed what helps to achieve more effective education.

Acknowledgements. This work was partially supported by the Slovak Research and Development Agency under the contract No. APVT-20-00710, the State programme of research and development "Establishing of Information Society" under the contract No. 1025/04, the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 3/5187/07 and the Scientific Grant Agency of Slovak Republic, grant No. VG1/0508/09.

References

[1] R. Albertoni and M. D. Martino. Semantic similarity of ontology instances tailored on the application context. In *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, volume 4275, pages 1020–1038. Springer Berlin/Heidelberg, 2006.

[2] A. Andrejko and M. Bieliková. Comparing instances of ontological concepts for personalized recommendation in large information spaces. *Computing and Informatics*, 28(4):427–450, 2009.

[3] I. Androutsopoulos, S. Kallonis, and V. Karkaletsis. Exploiting OWL ontologies in the multilingual generation of object descriptions. In *Proc. of the 10th European Workshop on Nat. Lang. Generation*, pages 150–155, Aberdeen, Scotland, 2005.

[4] M. Barla, M. Tvarožek, and M. Bieliková. Rule-based user characteristics acquisition from logs with semantics for personalized web-based systems. *Computing and Informatics*, 28(4):399–426, 2009.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[6] M. Bieliková and P. Návrat. Modeling and acquisition, processing and employing knowledge about user activities in the internet hyperspace (in slovak). In *Znalosti 2007: Proceedings of the 6th annual conference*, pages 368–371, Ostrava, Czech Republic, 2007.

[7] M. Bieliková and P. Návrat. Adaptive web-based portal for effective learning programming. *Communication & Cognition*, 42(1/2):75–88, 2009.

[8] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[9] P. Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110, 2001.

[10] P. Brusilovsky. Developing adaptive educational hypermedia systems: From design models to authoring tools. In *Authoring Tools for Advanced Technology Learning Environments*, pages 377–409. Dordrecht: Kluwer Academic Publishers, 2003.

[11] C. Callaway and T. Kuflik. Using a domain ontology to mediate between a user model and domain applications. In *Workshop on New Technologies for Personalized Information Access*, pages 13–22, Edinburgh, Scotland, UK, 2005.

[12] P. De Bra, A. Aerts, B. Berden, B. de Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash. AHA! The adaptive hypermedia architecture. In *ACM Conf. on Hypertext and Hypermedia*, pages 81–84, Nottingham, UK, 2003.

[13] R. Denaux, L. Aroyo, and V. Dimitrova. An approach for ontology-based elicitation of user models to enable personalization on the sem. web. In *Special interest tracks and posters of the 14th Int. Conf. on WWW*, pages 1170–1171, 2005.

[14] P. Dolog, N. Henze, and W. Nejdl. The personal reader: Personalizing and enriching learning resources using semantic web technologies. In *Proc. of the AH 2004*, pages 85–94. Springer Verlag, 2004.

[15] A. Formica and M. Missikoff. Concept similarity in symontos: An enterprise ontology management tool. *Computer Journal*, 45(6):583–594, 2002.

[16] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.

[17] P. Karampiperis and D. Sampson. Adaptive learning resources sequencing in educational hypermedia systems. *Educational Technology & Society*, 8:128–147, 2005.

[18] A. Kavcic. The role of user models in adaptive hypermedia systems. In *10th Mediterranean Electrotechnical Conference MEleCon 2000*, Lemesos, Cyprus, 2000.

[19] J. Kay. Stereotypes, student models and scrutability. *Lecture Notes in Computer Science*, 1839:19–30, 2000.

[20] J. Kay and A. Lum. Ontology-based user modeling for the semantic web. In *10th International Conference on User Modeling (UM'05): Workshop 8*, pages 11–19, Edinburgh, Scotland, 2005.

[21] T. Kirsten, A. Thor, and E. Rahm. Instance-based matching of large life science ontologies. In *Data Integration in the Life Sciences*, volume 4544, pages 172–187. Springer Berlin/Heidelberg, 2007.

[22] G.-J. M. Kruijff. Context-sensitive utterance planning for CCG. In *Proceedings of the 10th European Workshop on Natural Language Generation*, pages 83–90, Scotland, 2005.

[23] D. Milward and M. Beveridge. Ontology-based dialogue systems. In *18th International Joint Conference on Artificial Intelligence (IJCAI03): 3rd Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2003.

[24] P. Návrat, M. Bieliková, and V. Rozinajová. Methods and tools for acquiring and presenting information and knowledge in the web. In *International Conference on Computer Systems and Technologies – CompSysTech'2005*, Varna, Bulgaria, 2005.

[25] E. Rich. User modeling via stereotypes. In *Readings in intelligent user interfaces*, pages 329–342. Morgan Kaufmann Publishers Inc., 1998.

[26] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data Knowledge Engineering*, 25(1-2):161–197, 1998.

[27] M. Tvarožek, M. Barla, and M. Bieliková. Personalized presentation in web-based information systems. In *SOFSEM 2007*, Lecture Notes in Computer Science 4362, pages 682–692. Springer-Verlag, 2007.

Selected Papers by the Author

A. Andrejko, M. Bieliková. Comparing instances of ontological concepts for personalized recommendation in large information spaces. *Computing and Informatics*, 28(4): 427–450, 2009.

A. Andrejko, M. Bieliková. Personalized Comparing Instances of Domain Ontology Concepts. In Phivos Mylonas, Manolis Wallace, and Marios Angelides, editors, *Proceedings of SMAP 2008 – 3rd International Workshop on Semantic Media Adaptation and Personalization*, pages 76–81, Prague, Czech Republic, 2008. CS IEEE Press.

A. Andrejko, M. Bieliková. Investigating similarity of ontology instances and its causes. In *Artificial Neural Networks – ICANN 2008*, LNCS 5164, pages 1–10, Prague, Czech Republic, 2008. Springer.

M. Šimún, A. Andrejko, M. Bieliková. Maintenance of learner's characteristics by spreading a change. In *Learning to Live in the Knowledge Society*, volume 281 of *IFIP International Federation for Information Processing*, pages 223–226. Springer Boston, 2008.

P. Bartalos, M. Barla, G. Frivolt, M. Tvarožek, A. Andrejko, M. Bieliková, P. Návrat. Building ontological base for experimental evaluation of semantic web applications. In J. van Leeuwen, G.F. Italiano, W. van der Hoek, C. Meinel, H. Sack, and F. Plášil, editors, *SOFSEM 2007*, LNCS 4362, pages 682–692. Springer, 2007.

A. Andrejko, M. Barla, M. Bieliková. Ontology-based user modeling for web-based information systems. In Wita Wojtkowski, W. Gregory Wojtkowski, Jože Zupancic, Gabor Magyar, and Gabor Knapp, editors, *Advances in Information Systems Development, New Methods and Practice for the Networked Society*, volume 2, pages 457–468. Springer, New York, 2007.

A. Andrejko, M. Barla, M. Bieliková, M. Tvarožek. User characteristics acquisition from logs with semantics. In *ISIM 2007: 10th International Conf. on Information System Implementation and Modeling*, pages 103–110, 2007.

M. Šimún, A. Andrejko, M. Bieliková. Ontology-based models for personalized e-leraning environment. In *ICETA 2007: 5th Internation Conference on Emerging e-Learning Technologies and Applications*, pages 335–340. Elfa, Stará Lesná, Slovak Republic, 2007.

M. Bieliková, J. Kuruc, A. Andrejko. Learning Programming with Adaptive Web-Based Hypermedia System AHA!. In *ICETA 2005: 4th Internation Conference on Emerging e-Learning Technologies and Applications*, pages 251–256. Elfa, Košice, Slovak Republic, 2005.

A. Andrejko, M. Bieliková. Estimating similarity of the ontological concepts instances for the adaptive applications based on Semantic Web (in Slovak). In V. Snášel, editor, *Znalosti 2008: Proceedings of the 7th annual conference*, pages 30–41. STU, Bratislava, 2008.

M. Šimún, A. Andrejko, M. Bieliková. Initialization and actualization of the user model in job offers search on the Web (in Slovak). In *Znalosti 2007: Proceedings of the 6th annual conference*, pages 109–120. VŠB-Technická univerzita Ostrava, 2007.

A. Andrejko, M. Bieliková. Comparing Instances of the Ontological Concepts. In Pavol Návrat, Pavol Bartoš, Mária Bieliková, Ladislav Hluchý, and Peter Vojtáš, editors, *Tools for acquisition, organization and presenting of information and knowledge (2): Research project workshop*, pages 26–35, Horský hotel Poľana, Slovakia, 2007.

T. Klempa, A. Andrejko, M. Bieliková. Maintenance of user characteristics in the user model based on generating questions according to domain ontology concepts (in Slovak). In Peter Vojtáš, editor, *ITAT 2007: Informačné technológie – Aplikácie a Teória: Zborník príspevkov prezentovaných na pracovnom seminári ITAT Poľana*, pages 3–8, Poľana, Slovakia, 2007.

A. Andrejko, M. Bieliková. Estimating similarity of the ontological concepts instances for personalization purposes (in Slovak). In F. Babič and J. Paralič, editors, *WIKT 2007 Proceedings: 2nd Workshop on Intelligent and Knowledge oriented Technologies*, pages 46–49. Košice, 2008.

A. Andrejko, M. Barla, M. Bieliková, M. Tvarožek. Software tools for acquisition of user characteristics. In P. Vojtáš, T. Skopal, editors, *Datakon 2006, Proceedings of the Annual Database Conference*, pages 139–148. Brno, Czech Republic, 2006.

A. Andrejko, M. Barla, M. Tvarožek. Comparing ontological concepts to evaluate similarity. In Pavol Návrat, Pavol Bartoš, Mária Bieliková, Ladislav Hluchý, and Peter Vojtáš, editors, *Tools for Acguisition, Organisation and Presenting of Information and Knowledge: Research Project Workshop*, pages 71–78, Bystrá dolina, Nízke Tatry, 2006.

M. Tvarožek, M. Barla, M. Bieliková, V. Grlický, A. Andrejko, R. Filkorn, P. Bartalos. Presentation and Personalization of Information in the Semantic Web. In Pavol Návrat, Pavol Bartoš, Mária Bieliková, Ladislav Hluchý, and Peter Vojtáš, editors, *Tools for Acguisition, Organisation and Presenting of Information and Knowledge: Research Project Workshop*, pages 201–207, Bystrá dolina, Nízke Tatry, 2006.