D3.2.2 Usage-driven Change Discovery - Evaluation

Stephan Bloehdorn, Peter Haase and York Sure
(Institute AIFB, University of Karlsruhe)
with contributions from Daniel Ziegler

Abstract.
EU-IST Integrated Project (IP) IST-2003-506826 SEKT
Deliverable D3.2.2 (WP3.2)
In this deliverable we present the evaluation of our work performed in the task 'T3.2 Usage-driven Ontology Evolution'.
Keyword list: Usage Analysis, Ontology Evolution, Ontology Management

Document Id.	SEKT/2005/D3.2.2/v1.0
Project	SEKT EU-IST-2003-506826
Date	November 2, 2005
Distribution	public

SEKT Consortium

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-2003-506826.

British Telecommunications plc.
Orion 5/12, Adastral Park
Ipswich IP5 3RE
UK
Tel: +44 1473 609583, Fax: +44 1473 609832
Contact person: John Davies
E-mail: john.nj.davies@bt.com

Jozef Stefan Institute
Jamova 39
1000 Ljubljana
Slovenia
Tel: +386 1 4773 778, Fax: +386 1 4251 038
Contact person: Marko Grobelnik
E-mail: marko.grobelnik@ijs.si

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Tel: +44 114 222 1891, Fax: +44 114 222 1810
Contact person: Hamish Cunningham
E-mail: hamish@dcs.shef.ac.uk

Intelligent Software Components S.A.
Pedro de Valdivia, 10
28006 Madrid
Spain
Tel: +34 913 349 797, Fax: +49 34 913 349 799
Contact person: Richard Benjamins
E-mail: rbenjamins@isoco.com

Ontoprise GmbH
Amalienbadstr. 36
76227 Karlsruhe
Germany
Tel: +49 721 50980912, Fax: +49 721 50980911
Contact person: Hans-Peter Schnurr
E-mail: schnurr@ontoprise.de

Vrije Universiteit Amsterdam (VUA)
Department of Computer Sciences
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
Tel: +31 20 444 7731, Fax: +31 84 221 4294
Contact person: Frank van Harmelen
E-mail: frank.van.harmelen@cs.vu.nl

Empolis GmbH
Europaallee 10
67657 Kaiserslautern
Germany
Tel: +49 631 303 5540, Fax: +49 631 303 5507
Contact person: Ralph Traphöner
E-mail: ralph.traphoener@empolis.com

University of Karlsruhe, Institute AIFB
Englerstr. 28
D-76128 Karlsruhe
Germany
Tel: +49 721 608 6592, Fax: +49 721 608 6580
Contact person: York Sure
E-mail: sure@aifb.uni-karlsruhe.de

University of Innsbruck
Institute of Computer Science
Technikerstraße 13
6020 Innsbruck
Austria
Tel: +43 512 507 6475, Fax: +43 512 507 9872
Contact person: Jos de Bruijn
E-mail: jos.de-bruijn@deri.ie

Kea-pro GmbH
Tal
6464 Springen
Switzerland
Tel: +41 41 879 00, Fax: 41 41 879 00 13
Contact person: Tom Bösser
E-mail: tb@keapro.net

Sirma AI EAD, Ontotext Lab
135 Tsarigradsko Shose
Sofia 1784
Bulgaria
Tel: +359 2 9768 303, Fax: +359 2 9768 311
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Universitat Autonoma de Barcelona
Edifici B, Campus de la UAB
08193 Bellaterra (Cerdanyola del Vallès)
Barcelona
Spain
Tel: +34 93 581 22 35, Fax: +34 93 581 29 88
Contact person: Pompeu Casanovas Romeu
E-mail: pompeu.casanovas@uab.es

Executive Summary

The world is constantly changing, and so does required and available knowledge, e.g. stored in Digital Libraries. Knowledge workers heavily rely on the availability and accessibility of knowledge contained in such libraries. The sheer mass of knowledge available today, however, requires sophisticated support for searching and, often considered as equally important, personalization.

In SEKT we address these challenges by using ontologies. Ontologies by nature make implicit knowledge explicit, they describe relevant parts of the world and make them machine understandable and processable. To be effective, ontologies need to change possibly as fast as the parts of the world they describe

Change discovery aims at generating implicit requirements by inducing ontology changes from existing data. We here focus on *usage-driven* change discovery. Usage data is a very valuable source of contextual information, based on which the ontology can be modified in order to reflect changes in the real world,

In a precedent deliverable we have presented a framework for usage-driven ontology evolution along with a set of methods for change discovery. In this deliverable we evaluate this framework by applying it to actual use cases from the BT DL case study. The use cases include (1) usage-based pruning of a generic ontology to obtain a personalized ontology, and (2) extending an existing ontology with personalized extensions obtained via Ontology Learning techniques. The evaluation results show the usefulness of applying usage data for the task of ontology evolution.

Contents

Chapter 1

Introduction

1.1 Motivation

The world is constantly changing, and so does required and available knowledge, as e.g. stored in Digital Libraries. Knowledge workers heavily rely on the availability and accessability of knowledge contained in such libraries. The sheer mass of knowledge available today, however, requires sophisticated support for searching and, often considered as equally important, personalization.

In SEKT we address these challenges by means of ontology and metadata technologies. Ontologies by nature make implicit knowledge explicit, they describe relevant parts of the world and make them machine understandable and processable. To be effective, ontologies need to change possibly as fast as the parts of the world they describe.

For the understanding of this deliverable it is important to distinguish between *change capturing* and *change discovery*, which we define in the following definitions:

Definition 1 (Change Capturing) *The task of* change capturing *can be defined as the generation of ontology changes from explicit and implicit requirements.*

Here, explicit requirements are generated, for example, by ontology engineers who want to adapt the ontology to new requirements or by the end-users who provide the explicit feedback about the usability of ontology entities. The changes resulting from this kind of requirements are called *top-down changes*. Implicit requirements leading to so-called *bottom-up changes* are reflected in the behavior of the system and can be induced by applying *change discovery* methods.

Definition 2 (Change Discovery) Change discovery *aims at generating implicit requirements by inducing ontology changes from existing data.*

In [Sto04] three types of change discovery are defined:

1. structure-driven,
2. usage-driven and
3. data-driven.

While *structure-driven changes* can be deduced from the ontology structure itself, *usage-driven changes* result from the usage patterns created over a period time. *Data-driven changes* are generated by modifications to the underlying data, such as text documents or a database, representing the knowledge modelled by an ontology.

In this deliverable, we focus on the *usage-driven* change discovery. Usage data is a very valuable source of contextual information, based on which the ontology can be modified in order to reflect changes in the real world, changes in user's requirements, drawbacks in the initial design, to incorporate additional functionality or to allow for incremental improvement.

In a precedent deliverable we have presented a framework for usage-driven ontology evolution along with a set of methods for change discovery. In this deliverable we evaluate this framework by applying it to actual use cases from the BT DL case study. The use cases include (1) usage-based pruning of a generic ontology to obtain a personalized ontology, (2) extending an existing ontology with personalized extensions obtained via Ontology Learning techniques, and (3) generating information space specific ontologies based on usage data.

1.2 Application Scenario

In this section we present a typical application scenario to motivate and illustrate our approach. As part of the EU IST SEKT project[1] we realize with other SEKT partners several case studies including a case study at British Telecom (BT). In the BT Digital Library so-called information spaces have been created for domains known to be of interest to people in the company to structure the contents of journals in the library. One of the key elements of the case study is to use ontologies to enhance the knowledge access to the Digital Library. Interests of people change over time, as does the content of the digital library. The ontologies need to adapt to those changes in order to stay current and useful.

Figure 1.1 offers an overview of the main building blocks to support the evolution of ontologies in the Digital Library, which we explain in more detail in the following.

Users of the Digital Library interact with the *knowledge portal* as the user interface. The knowledge portal allows the user to search the library's contents as it presents the content in an organized way. The knowledge portal may also provide the user with information in a proactive manner, e.g. by alerts, notification, etc. The typical user primarily

[1] http://www.sekt-project.com

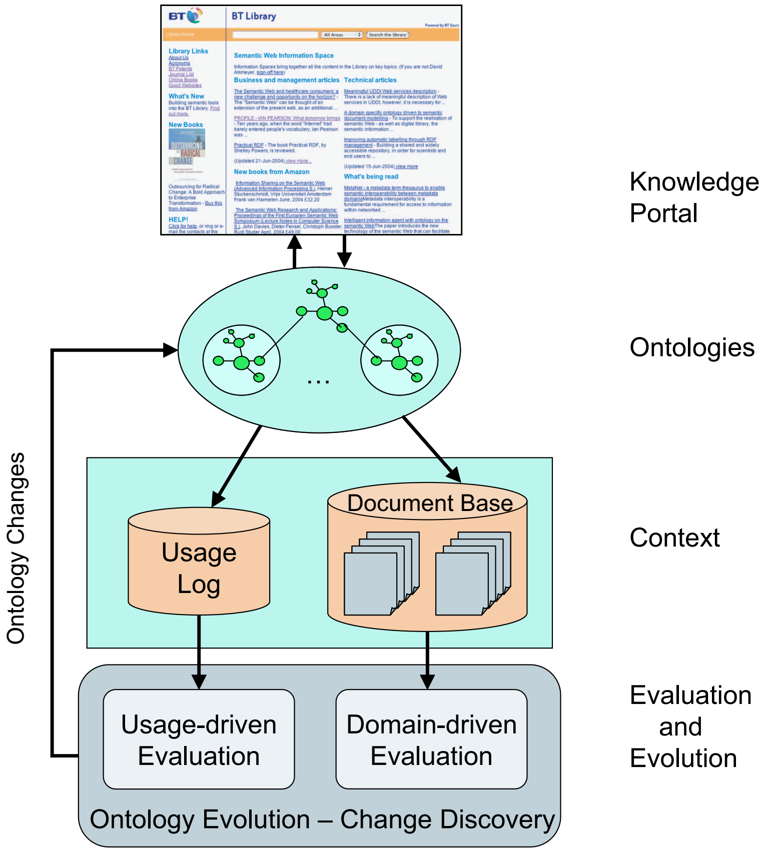

Figure 1.1: Logical Architecture

consumes knowledge from the Digital Library. She uses the Digital Library to fulfil a particular information need. However, an advanced user may also contribute to the Digital Library, either by contributing content or by organizing the existing content, providing metadata, etc.

Ontologies are the basis for rich, semantic descriptions of the content in the Digital Library. Here we can identify two main modules of the ontology:

The *application ontology* describes different generic aspects of bibliographic metadata (such as *author* or *creation data*) and is valid across various bibliographic sources.

Domain ontologies describe aspects that are specific to particular domains and are used as a conceptual backbone for structuring the domain information provided in the information spaces. Such a domain ontology typically comprises conceptual relations, such as a topic hierarchy, but also richer taxonomic and non-taxonomic relations.

Personalisation of ontologies further tailors the interaction of the user with the knowledge portal to a particular user. Thus a view on the domain ontologies that reflect the interests of the user at a certain time can be represented. This allows to better deal with the large size of the data by selecting the relevant parts of the ontology.

The ontologies are used for various purposes: First of all, the documents in the document base are annotated and classified according to the ontology. This ontological metadata can then be exploited for advanced knowledge access, including navigation, browsing, and semantic searches. Finally, the ontology can be used for the visualization of results, e.g. for displaying the relationships between information objects.

While the application ontology can be assumed to be fairly static and applicable across information spaces, the domain ontologies (including personalized ontologies) must be continuously adapted to the changing domain and user needs.

The *document base* comprises a corpus of documents. The content of the document base typically is not static, but changes over time: New documents come in, but also documents may be removed from the document base. The document base holds documents from one or more domains. *Information Spaces* are the logical units to organize a collection of documents according to domains.

The interaction of the knowledge worker with the knowledge portal is recorded in a *usage log*. It is of particular interest how the ontology has been used in the interaction, i.e. which elements have been queried, which paths have been navigated, etc. By tracking users' interactions with the application in a log file, it is possible to collect useful information that can be used to assess what the main interests of the users are. In this way, we are able to obtain implicit feedback and to extract needs for changes to the ontology to improve the interaction with the application.

Both the usage log information as well as the document corpus for a particular domain establish a *context* for the ontology. This contextual information is a valuable input for the *evaluation* of the ontology. Specifically, we can define the value of the ontology as how good it reflects the interests of the user and allows the user to obtain the relevant information, and how well it reflects the domain of the document corpus.

Based on the evaluation of ontologies, we can guide the *evolution* of the ontology by discovering and applying potentially useful changes that increase the value of the ontology, i.e. we generate changes to the ontology to improve the interaction with the Digital Library. While the recommendations for ontology changes are generated in an automated manner, they typically will be approved by a knowledge engineer before the actual application.

1.3 Related Work

In the preceding deliverable [HS05b] we have already presented an extensive overview on relevant related work. At this point, we would just like to discuss two recent works that deserve attention:

In [SMB04] the authors present a framework for contextualized information access based on a user profile that reveals long-term user interests and trends. The problem is modeled in the context of our client-side Web agent ARCH (Adaptive Retrieval based on Concept Hierarchies). In ARCH, the user profiles are generated using an unsupervised document clustering technique. These profiles, in turn, are used to automatically learn the semantic context of user's information need from a domain-specific concept hierarchy. Experimental results show that implicit measures of user interests, combined with the semantic knowledge embedded in a concept hierarchy, can be used effectively to infer the user context and improve the results of information retrieval.

In [DM05] the authors argue that it is essential to take into account the semantic knowledge about the underlying domain for the task of personalized information access. Without such semantic knowledge, personalization systems cannot recommend different types of complex objects based in their underlying properties and attributes, nor can these systems possess the ability to automatically explain or reason about the user models or user recommendations. The integration of semantic knowledge is, in fact, the primary challenge for the next generation of personalization systems.

We see these results as a confirmation of our work on using user profile information to generate appropriate ontologies for improved information access.

1.4 Overview of the Deliverable

In the remainder of this deliverable we will present the approach to usage-driven ontology evolution for the two specific use cases in Chapter 2. In Chapter 3 we will present the evaluation the approaches, including a description of the evaluation setting, evaluation measures and evaluation results. We conclude with an outlook to future work in Chapter 4.

Chapter 2

Approach

2.1 Personal Ontologies in the BT Digital Library Case Study

The use of personal ontologies is one particular instrument to realize personalized information access to a Digital Library. Personal ontologies allow to browse the content of the library and visualize results in an adequate manner by employing knowledge structures that are relavant and comprehensible for the user. The need and usefulness of such personal ontologies – as opposed to generic ontologies – is immediately evident if one considers that

1. generic topic ontologies such as ABI and Inspec[1] consist of tens of thousands of concepts and are very broad in scope, while only a small subset of the ontology is actually relevant to the user,

2. despite their broad scope and size, these generic ontologies ary typically limited in depth, such that important concepts that are relevant to the user are not modelled.

These two observations are the motivation for two use cases for usage-driven ontology evolution to (1) prune an existing ontology to obtain a personalized ontology, and (2) learn user-specific extensions of a personalized ontology, presented in Sections 2.5 and 2.6 respectively. Additionally we can note that the generation of specialized ontologies not only makes sense for personalized ontologies, but also for dedicated ontologies for certain sets of documents such as information spaces, which reflect the interests of a larger set of users.

Figure 2.1 shows how the components to fit into the architecture of the BT DL case study. Figure 2.2 shows the usage of the personalized ontologies in the user interface of the BT DL.

[1] `http://www.iee.org/publish/inspec/`

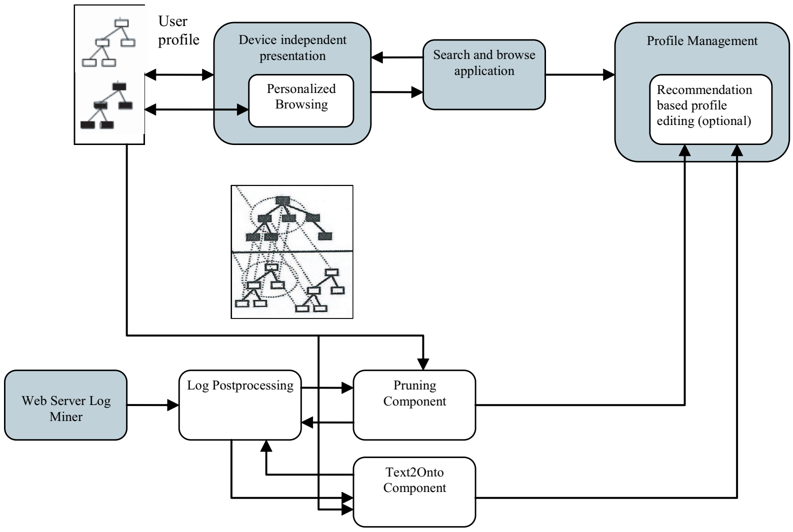

Figure 2.1: Usage-driven ontology evolution process in the BT DL

2.2 Ontology Evaluation for Ontology Evolution

In this section we provide a very brief review of our approach to ontology evolution based on ontology evolution, as we have first introduced in [HS05b] and then integrated into our framework for incremental ontology evolution in [HS05a]. For a full account, we refer the reader to the above two deliverables.

Ontology model In continuation of our work in D3.1.1 we base our work on the OWL DL ontology model. As usual in description logics, an OWL ontology is built over a vocabulary that consists a set of concept names N_C, sets of abstract and concrete individual names N_{I_a} and N_{I_c}, respectively, and sets of abstract and concrete role names N_{R_a} and N_{R_c}, respectively. The set of OWL DL *concepts* is defined by a set of syntactic rules, an ontology then is a finite set of axioms over these concepts. In the following, we denote the set of all ontology elements, i.e. both axioms and symbol names, with N. We denote the set of all possible ontologies with $\mathcal{O}$.

Context Model Our ontology model so far describes the actual state of an ontology as an isolated entity. Once we enter the more dynamic scenario of ontology evolution, it makes sense to consider contextual information about the ontology. The term "context" has many different connotations depending on the field it is being used in. In general, the context of an entity includes the circumstances and conditions which "surround" it. [Dey01] has defined context as: *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered*

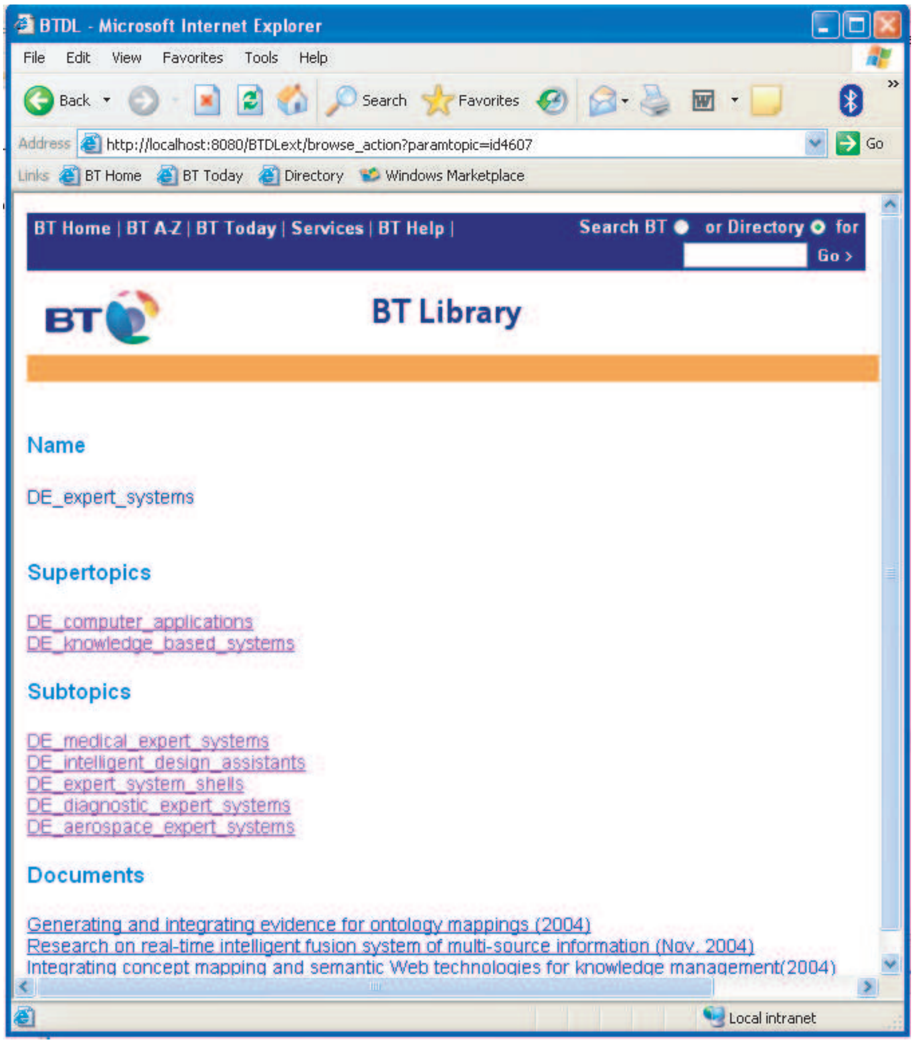

Figure 2.2: Screenshot

relevant to the interaction between a user and an application, including the user and applications themselves. Applied to ontologies this means that we consider any information that is external to the ontology itself, but relevant to the interaction between the user and the ontology-based application. Typical examples of contextual information include – as we will later elaborate – usage log information, provenance, information about trust, confidence and certainty, etc.

To capture such contextual information about ontologies, we introduce the notion of *ontology rating annotations*, which are used to relate the context to the elements of the ontology:

Definition 3 *Let N denote the set of all possible ontology elements and $\mathcal{X}$ be a suitable representation of a context space, then an ontology rating annotation is a partial function $r : N \to \mathcal{X}$.*

Please note that the ontology elements to be rated – according to the definition of the ontology model – can be either axioms or entities. The context space $\mathcal{X}$ is kept as general as possible in this definition, to allow to attach essentially arbitrary information external to the ontology. We will instantiate $\mathcal{X}$ for the specific contexts we consider.

Ontology Evaluation Good ontologies are ontologies that serve their purpose. In order to be able to define what a "good" ontology for a particular context is, we need to be able to measure the quality of the ontology with respect to a given set of criteria. For our framework, we do not commit to a certain set of evaluation methodologies, but rather allow the user of this framework to choose the evaluation approach she deems best fitting for her task.

The relationship between intended models and specification is being captured by the context, as described in the former section, and is measured by an *ontology evaluation function*.

Definition 4 *Let $\mathcal{O}$ be the set of all possible ontologies, then an ontology evaluation function e is a function $e : \mathcal{O} \rightarrow [0, 1]$.*

Effectively, the evaluation function provides a total order over the space of possible ontologies and thus allows to compare given ontologies. Here it is important to note that the evaluation function can take the rating annotations into account and thus provides an evaluation measure with regards to a given context. The intuitive reading of the evaluation function is that a value of 1 indicates the "perfect" ontology, whereas a value of 0 is the "worst case".

By using the unit interval for the representation of the value of an ontology, we obtain the immediate benefit of being able to combine different quality criteria, e.g. using a weighted average of different ontology evaluation functions.

Ontology Evolution Ontology evolution is timely adaptation of the ontology to changes and the consistent management of these changes. In particular, we need to account for changes in the context of the ontology, i.e. to evolve the ontology with its context. To operationalize this, we first formalize the notion of ontology changes. Based on the ontology evaluation function we are then able to determine whether a particular change leads to an improved ontology. The actual challenge then is the discovery of potentially useful changes.

Definition 5 (Ontology Change Operation) *An* ontology change operation oco *is a function*

$$oco : \mathcal{O} \rightarrow \mathcal{O}$$

With OCO we denote the set of all possible ontology change operations. For the ontology model defined above, we allow the atomic change operations of adding and removing axioms, which we denote with α^+ and α^-, respectively. Complex ontology change operations can be expressed as a sequence of atomic ontology change operations. The semantics of the sequence is the chaining of the corresponding functions: For some atomic change operations $\text{oco}_1, ..., \text{oco}_n$ we can define $\text{oco}_{\text{complex}} = \text{oco}_n \circ ... \circ \text{oco}_1 = \text{oco}_n(...\text{oco}_1)$.

Change Discovery Based on the ontology evaluation function, we can now measure whether a particular change to an ontology leads to an "improvement" of the ontology for the given context. As this context changes over time, we can regard ontology evolution as the adaptation to the changing context by discovering and applying changes to the ontology. Essentially, the goal is to discover changes that lead to a maximized evaluation function, i.e. the ideal ontology for the particular context:

Definition 6 *For a given ontology O and an evaluation function e, we can define the problem of change discovery as an optimization problem:*

$$max_{oco \in OCO}\ e(oco(O))$$

Having the problem stated as an optimization problem opens the door to applying established optimization techniques to find the "best" ontology with respect to the evaluation function. In general, it will be hard to determine the optimal ontology that maximizes the evaluation function, as one theoretically would need to search the entire space of possible consistent ontologies. However, in most cases it is not necessary to prove the optimality of an obtained solution. Instead it is possible to exploit heuristics-based techniques to obtain a "fairly" optimal ontology.

2.3 Representation of Usage-Context

The intention of modelling usage context is to capture the users' behavioral patterns, which can in turn be used to assess the effectiveness in the interaction with the ontology, to identify important parts, but also weaknesses of the ontology. We here use the rating annotations to indicate the importance of particular elements. The methods presented in this section have - to a large extent - already been presented in [HS05b]. We here extend them and apply them to the use cases of the BT DL.

Generally, we can distinguish between *explicit* and *implicit* user feedback from usage information. We talk about explicit feedback if we allow that a user (i) can express in a more fine-grained way how important a certain ontology element is for him, and that he (ii) can explicitly express negative ratings for ontology elements that she does not want to be part of his ontology.

Example 1 (Explicit Usage Rating) *We use an explicit rating, called the membership-rating* $r^m : N \rightarrow \{-1, 0, +1\}$*, for which (i) all symbols and axioms the user actually wants to be part of the ontology have rating* $+1$*, and (ii) all symbols and axioms not actually part of the ontology can be explicitly marked by the user with a rating* -1*. Finally,* 0 *indicates an unrated element.*

Implicit feedback we can obtain from log information that indirectly indicate the importance of ontology elements based on how they have been used.

Example 2 (Implicit Usage Rating) *We use an implicit, usage-based rating called* $r^u : N \rightarrow \mathbb{N}$*, which indicates the relevance of the elements based on how they have been used, e.g. counts the number of queries issued by the user and instances in her knowledge base that reference a given symbol name.*

We consider two implicit usage rating annotations:

- $r^u_{queries}$ *annotates the ontology elements with the number of queries that have referenced the particular ontology elements,*
- $r^u_{browsing}$ *annotates the ontology elements with the number of browsing activities for the particular ontology elements, e.g. the number of times a link has been followed in the case of a subtopicof relation.*
- $r^u_{instances}$ *annotates the ontology elements with the number of instances that are classified under the particular ontology elements.*

The two rating annotations capture two important and typical dimensions of usage, one with respect to the content (how the concepts are used to classify instances), and one with respect to the usage by the end users, i.e. which concepts were actually queried. This information is available in a wide range of application scenarios. Of course, in specific scenarios further information may be available and thus additional rating annotations can be defined.

User Profiles A further input may be the already existing representation of an explicit user profile, e.g. generated by methods developed in [BGM05]. These user profiles are represented as shown in Figure 2.3. In this case, the `degreeOfInterest` can be used as an implicit user reating for the respective concept. It further allows to distinguish between a "general interest" (`hasInterest`) and a "current interest" (`hasCurrentInterest`).

2.4 Representation of Personal Ontologies

Personal ontologies introduce the possibility for users to define their own view and extension of the general, global ontology (e.g. Inspec and ABI topic hierarchies). We primarily

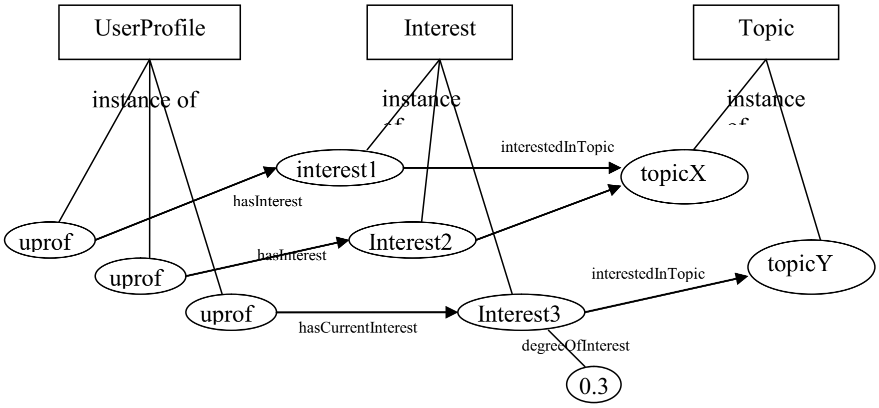

Figure 2.3: Representation of User Profiles

aim at enabling the user to extend the existing structure by adding new topic instances and subTopicOf-relations and to exclude elements of the common structure. The representation of personal ontologies is shown in Figure 2.4. In order to keep the structure as clear and concise as possible we do not store the personal extensions in the common ontology but introduce a personal ontology for each user. This has the additional advantage that simultaneous updates on data objects do not occur. The ontologies are referenced as first class objects in the common ontology. Additionally, a new property hasOntology with the class `UserProfile` as its domain and the class `owl:Ontology` as its range is introduced. The entities of the common ontology are included by importing the common ontology into the personal ontologies. Consequently, the set of topics in the common ontology will be a subset of the set of topics in the personal ontology. This implies that the Interest instances must be moved to the personal ontology.

Up to this point, it is only possible to declare additional elements in the personal ontology. In contrast, the focus of ontology pruning lies on reducing the complexity of the topic hierarchy by pruning it. Therefore, we additionally introduce a Boolean property `invisible` for the personal ontologies that is to be instantiated with the value `true` if a property is removed from a personal ontology. Of course, this is only done for Topic instances of the common ontology. Personal instances are completely removed.

Summarized, the derivation of the personal structure of a user can be obtained from the common and personal ontology in a two-step operation: first, the common ontology is extended by the Topic instances and additional `subTopicOf` properties of the personal ontology. In the second step the elements which are not supposed to be part of the personal structure are removed by retrieving the ”invisible” properties in the personal ontology.

The chosen representation does not allow to mark `subTopicOf` properties as being not existent in a personal profile (since, as an ObjectProperty, they cannot have properties by themselves). Instead, this is done indirectly by defining that a `subTopicOf` property is considered to be part of the personal structure if both instances it links are part of it

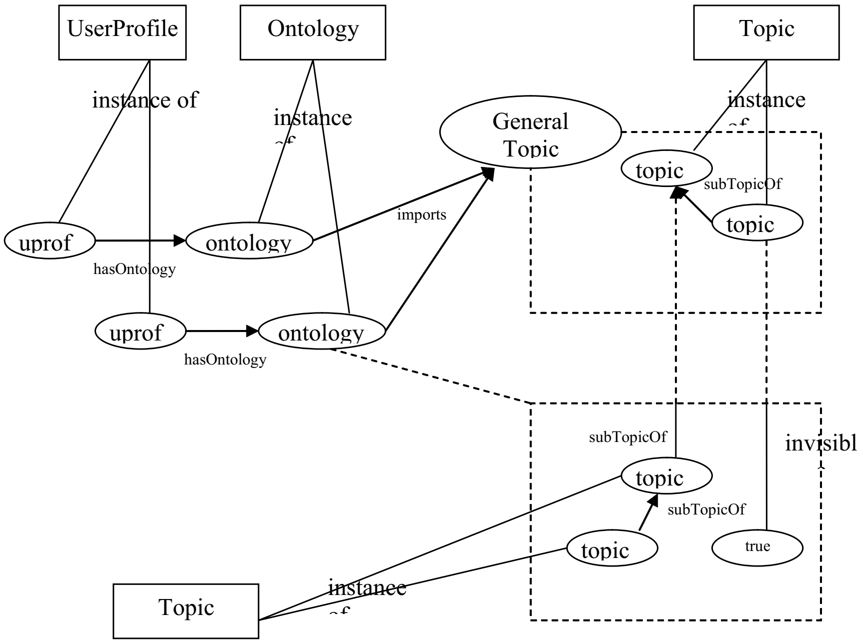

Figure 2.4: Representation of Personal Ontologies

(i.e. not marked as invisible). Consequently, the removal of a `subTopicOf` relation is comparatively complicated: One of the two instances (we use the parent instance) is marked as invisible, then a new Topic instance is created in the personal profile and which replicates the features of the removed instance (with the exception of the respective subTopicOf relation, of course).

2.5 Usecase 1: Usage-driven Ontology Pruning

This use case is concerned with the generation of a personalized ontology from a large ontology – typically a topic ontology – by means of pruning and adapting the ontology for the user needs for the task of browsing-based information access. The goal of this pruning is to improve the ontology to increase the effectiveness in terms of the costs to access documents via the browsing.

2.5.1 Evaluation Function

The evaluation function considers the browsing behaviour of previous browsing sessions and assumes these to be a typical pattern. The underlying idea is that a user has an approximate vision of how to approach the information resource he is looking for, or with respect to the model, a vision of the path he intends to go. The overall usage of the topic structure over a period of time can consequently be seen as a set of paths. If $r^u_{browsing}(subtopic(i, j))$ is the number of times a user browses from topic i to topic j, then the overall search time TST is the decisions (=nodes on the path) $r^u_{browsing}(subtopic(i, j))$ multiplied with the time one decision takes. The evaluation function estimates how much time it takes a user to browse to the needed documents is affected by changes to the structure. If a path is interrupted because a link has been removed (indicated by the binary variable a_{ij}), we assume that the user feels forced to search by keywords or other means with a punishment time ”RemovalCosts” that must be derived from statistical data. In the proposed evaluation function we assume a linear relation between the number of subtopics of a topic i , $breadth(i)$ (the number of subtopics of topic i), which is expressed in the factor bf.

The time it takes to navigate from topic i to j can then be calculated as:

$$\textit{browsingTime}(i, j, a_{ij})) = a_{ij} * bf * breadth(i) + (1 - a_{ij}) * RemovalCosts$$

Based on this formula, we can then calculate the total search time TST for a given ontology O based on the usage information with the following formula:

$$TST(O) = \sum_{subtopic(i,j) \in O} r^u_{browsing}(subtopic(i, j)) * \textit{browsingTime}(i, j, a_{ij})$$

The value of an ontology can be considered as the inverse to the cost of the ontology, as with the following evaluation function

$$e_{TST}(O) = \frac{1}{TST(O)}$$

The goal of the pruning process therefore is to discover changes that lead to improvements of the ontology in terms of this evaluation function. In the following we present such a process based on heuristics.

2.5.2 Pruning Process

A general overview of the control flow of the pruning process is shown in Figure 2.5. The pruning component is triggered in regular time intervals.

The pruning algorithms – as explained in the following – iterate over the ontology structure and check in each node if a change of the structure (e.g. by merging leads) to a local improvement of the ontology according to the evaluation function. Merging is considered as a change that does not change the structure semantically as much as the removal of subtopicOf-relations. Therefore, the merging of subtopics is preferred over the removal of subTopicOf-relations. Consequently, the algorithm starts by trying to merge subtopics. Merging groups of subtopics with different sets of parents would mean to add

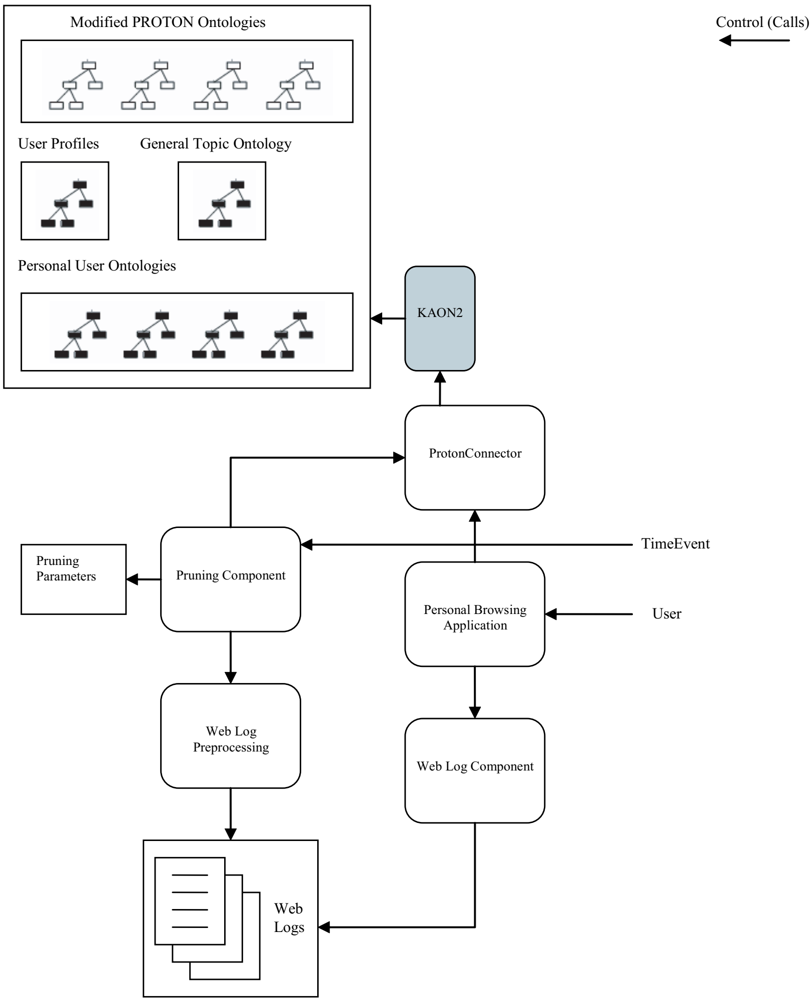

Figure 2.5: Overview of the Pruning Process

or remove subTopicOf-relations implicitly. In order to avoid that,the merge-operation is restricted to groups of subtopics where all members have identical sets of parents (called congruent topics). Only after removing subTopicOf-relations explicitly, we check if new merges have become possible.

The algorithm ImproveStructure (c.f. Algorithm 1) represents the main thread that controls the overall workflow of pruning the topic structure. For the single steps of its processing it uses various auxiliary algorithms. As parameters the algorithm receives a user profile UP modelling a personalized topic structure and a rating annotation each subtopicOf-relation with the usage intensity (i.e. the number of times the link is used for browsing downwards over the chosen time span) and each topic t with the sum of the usage intensities of the subTopicOf-relations where t is the subtopic. In a cyclic process it then checks first if there exist topics that can be merged having the identical set of parents: The function $BuildSetOfContruentTopics$ computes the maximal sets of topics which have identical parents, which is a defined prerequisite for two topics to be merged. Second the algorithm calculates the consequences to the evaluation function and executes the changes if the effect is positive. Third, it checks if the removal of subTopicOf-relations leads to an improvement (Algorithm "PositiveValue") and finally it executes favourable removals. Forth, it checks whether the removals have consequences on the sets of topics that can be merged. If this is the case, the cycle is repeated. Otherwise, the algorithm terminates.

Algorithm 1 Algorithm for Evolution of Topic Ontology

Require: A valid user profile UP,$r^u_{browsing}$

```
procedure IMPROVE(UserProfile UP, r^u_browsing )
    SCT := ∅
    SCTnew := BuildSetOfCongruentTopics(UP)
    while SCT ≠ SCTnew do
        SCT := SCTnew
        for all s ∈ SCT do
            BestMerge(s, UP, r^u_browsing, SCT)
        end for
        for all subTopicOf-relation r ∈ UP do
            if PositiveEffect(r, UP, r^u_browsing) = false then
                Remove(r, UP, continue)
            end if
        end for
        SCTnew := BuildSetOfCongruentTopics(UP)
    end while
end procedure
```

The algorithm BestMerge (c.f. Algorithm 2) receives a set of congruent topics as parameter and calculates the most beneficial subsets of the topics to be merged (if there is such a subset). Therefore it uses the rating annotations to the subTopicOf-relations. The positive effect of the removal of one topic by merging it with another on the estimated decision time in the parent topics is calculated. This is equal for all topics in the set. Now, the algorithm starts calculating the negative effect on the decision time for each pair in the set of congruent topics. It then selects the pair with the best (lowest) value and calculates the overall effect of this merge Benefits - Costs. If this value is positive it combines the two elements and calculates the effect of merging this new topic with all remaining members of the set. Again, it chooses the pair with the lowest value, calculates the net total effect Benefit - Costs as long as the effects are positive. This is repeated until there is no beneficial combination left.

The function in Algorithm 3 if a subtopic-relation j has a positive net effect on browsing structure according to evaluation function. It compares the estimated profit from removing a subTopicOf-relation j to the estimated costs. The profit arises from the fact that the decision times in the supertopic are assumed to decrease due to the decreased

Algorithm 2 Algorithm BestMerge

Require: Set of congruent topics SCT, user profile UP, rating annotation $r^u_{browsing}$

procedure BESTMERGE(*Set* $S, UP, r^u_{browsing}$, *Set* SCT)
    $Profit := 0$
    for all topics $p \in supertopics(S)$ **do**
        $Profit := Profit + r^u_{browsing}(p) * bf$
    end for
    Map ProposedMerges = new Map
    for all topics $t \in S$ **do**
        add new key/value pair to ProposedMerges with t as key and {t} as value
    end for
    Set Pairs = $\{(j,k)|j,k \in Topics, uri(j) < uri(k)\}$
    $MapCosts = newMap$
    for all pair $p = (j,k) \in pairs$ **do**
        $Costs(p) := (|\mathit{subtopics}(j)| * r^u_{browsing}(k) + |\mathit{subtopics}(k)| * r^u_{browsing}(j)) * bf)$
    end for
    Map Members = new Map
    for all pair $p - (j,k) \in pairs$ **do**
        add j and k to Members(p)
    end for
    loop
        $currPair := p \in Pairs : Costs(p) < Costs(q) forallq \neq p \in Pairs$
        if $Costs(currPair) \geq Profit$ **then**
            leave LOOP
        else
            $currTopic = newTopic$
            add $currTopic$ to $AvailableTopics$
            $\mathit{Children}(currTopic) := \mathit{Children}(j) + \mathit{Children}(k)$
            $r^u_{browsing}(currTopic) := r^u_{browsing}(j) + r^u_{browsing}(k)$
            $ProposedMerges(currTopic) := Members(p)$
            for all $p \neq currPair \in Pairs$ **do**
                if $Members(p) \cap Members(currPair) \neq \emptyset$ **then**
                    remove p from $Pairs$
                end if
                for all topic $m \in Members(currPair)$ **do**
                    remove m from $AvailableTopics$
                    if AvailableTopics $currTopic = \emptyset$ **then**
                        leave LOOP
                    end if
                    for all topics $m \in AvailableTopics \setminus \{currTopic\}$ **do**
                        add $(m, currTopic)$ to $Pairs$
                        $Members((m, currTopic)) := ProposedMerges(currTopic) \cup ProposedMerges(m)$
                        $Costs((m, currTopic)) := \mathit{Children}(m) * r^u_{browsing}(currTopic) + \mathit{Children}(currTopic) * r^u_{browsing}(m)$
                    end for
                end for
            end for
        end if
    end loop
end procedure

complexity. Unlike in the calculation of the profit of merging topics this holds only for the browsing activities which do not aim at the subTopic-relation which is removed. The calculation of the costs is based on the assumption that a user intending to use the subTopicOf-relation to reach a document will be forced to use less convenient and more time-consuming ways to reach the document he is looking for. This is expressed as a punishment time $RemovalCosts$ which can be interpreted as the average extra time it takes him to find the document.

Algorithm 3 PositiveEffect

Require: An existing subTopicOf-relation in the concerned user profile UP, ,$r^u_{browsing}$

```
function PositiveEffect(subTopicOf-relation j, UserProfile UP, ,r^u_browsing )
    Topicsuper := supertopic(j)
    Profit := (r^u_browsing(super) − r^u_browsing(j)) ∗ bf)
    Costs := r^u_browsing(j) ∗ RemovalCosts
    if Profit − Costs > 0 then
        return TRUE
    else
        return FALSE
    end if
end function
```

2.6 Usecase 2: Extending a Personal Ontology

With this use case we address the issue of discovering useful *extensions* to an existing ontology to further personalize it. While the prior use case was concerned about removing irrelevant parts from an ontology, this use case is concerned about adding potentially relevant parts. The obvious question is where these relevant parts should be obtained from. We here rely on the data-driven change discovery methods of Text2Onto developed in [VS05a] and [VS05b] and integrate the usage-driven approach to tailor the learning process. The Text2Onto framework has been designed for ontology learning based on data-driven change discovery from textual resources like HTML-pages, PDF documents or txt-files. The output of the framework is recommendations for ontological axioms like concept definitions, concept instantiations, role definitions and subconcept-of-relations etc.

We extend the ontology learning process in Text2Onto as shown in the Figure 2.6. The main idea of the approach is to tailor the learning by selecting “the right” subset of documents based on the usage-data:

1. As a first step, we apply the analysis of the usage-data to identify a subset of documents which is representative for the user’s interest.

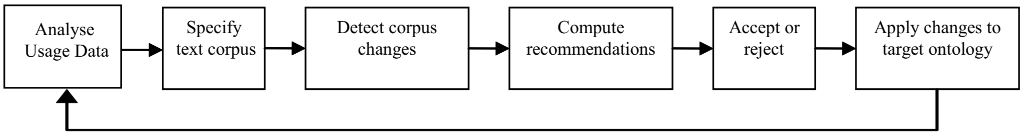

Figure 2.6: Extended Text2Onto Learning Process

2. The result of the first step is used to specify the text corpus as an input for the learning algorithms.

3. In cycles, the data change discovery methods are applied to the corpus. The changes are stored.

4. Ontology learning algorithms are applied to the changes according to user settings. A set of instances is created each representing an ontological entity or statement in connection with the text fragment that indicates its existence and a value that expresses the quality of the proposal (the certainty).

5. Human interaction can take place for approving or rejecting the proposals. Instead of a direct interaction there is also the possibility to execute a certain strategy in the form of a function that may draw external background knowledge into concern.

6. Last, the approved changes can be translated in various ontology languages.

The evaluation function that is applied to generate the learned ontologies using Text2Onto has been described in [HS05a]. For a full account we refer the reader to this deliverable. Its main intuition aims at capturing the most certain information for a given domain, while maintaining a consistent knowledge base to allow for meaningful query answering.

Chapter 3

Evaluation

In this chapter we present evaluation results for the approaches to ontology evolution based on the two use cases described in the previous chapter.

3.1 Evaluation of Usage-Driven Ontology Pruning

Evaluation Setting In the evaluation we assess whether the described pruning algorithm is able to improve personalized browsing structures of users. The changes which are computed by the algorithm are based on structural parameters like breadth, depth and the user behaviour. This can provide information about the importance of existing entities in the structure, but gives only limited evidence of how meaningful the existing entities are to the user. Therefore, the principal question is whether the proposed changes are accepted by the users, i.e. the users have the impression that the structure is more concise after the pruning.

For the evaluation, we focus on a group of five regular users of the BT Digital Library. For each of these users, a profile is created as described in Section 2.3. Since the current implementation of the BT Digital Library does not include the ability to browse on a topic structure, the user profiles are implemented in a prototype for the experiment: First, the set of documents is retrieved which the users have accessed between May 2004 and November 2005. Second, an initial profile is created for each user which includes all topics to which the documents stand in a ”hasSubject” relation.

A subset of the set of documents of each user is chosen. The users get the task to find these documents by browsing the topic structure. Based on these browsing activities the rating annotations are created. Then the pruning algorithm is executed on the initial profile.

Evaluation Measures For the evaluation, we rely on a system evaluation and a user evaluation based on an interview. The goal of the system evaluation is to determine whether an actual effect in terms of search time can be measured, whereas the user evaluation based on an interview aims at the subjective experience of the user.

System evaluation. The essential question for an evaluation of the presented approach is whether the recommended changes are suitable for reducing the total search time. Therefore, a first evaluation measure is the number of all successful accesses to documents divided by the sum of the durations of all user sessions. A successful access is defined as an access to the full text of an article or the creation of a bookmark to the article. The duration of a user session is defined as the time between the first and last action that is logged.

User evaluation based on an interview: A subset of the changes is used in the interview. The users are shown one change after the other and asked if they would approve of the proposed changes. They can choose among yes, no, and ”Don’t know, cannot see the use for me”. If a user selects ”no” he is asked for the reason. He can choose among different answers depending on the type of change operation. It is possible to select more than one answer:

Merge:

- ”Because the topics proposed for merge do not fit together.”
- ”Because the super topic is too narrow after the merge.”
- ”Because the merged topic is too broad after the merge.”

Other-Grouping:

- ”Because some of the topics grouped under ”Others” are still useful for me.”
- ”Because the topic is too narrow after the merge.”

Pulling-Up:

- ”Because the topic does not fit into the set of topics it is pulled up into.”
- ”Because the upper topic gets too broad by adding further topics.”
- ”Because the lower topic gets too narrow by removing further topics.”

Results The above described experiment is currently being performed, but not completed yet. The final results will be available in January and are to be included with the final version of this deliverable as an update to this version.

3.2 Evaluation of Extending Personalized Ontologies

Evaluation Setting To evaluate the approach of usage-driven extensions to a personalized ontology, we have performed a user study with users of the Digital Library. The goal of this user study was to evaluate whether a personalization of the process of extending an ontology is beneficial. In the experiment, we focused on one existing information space of the BT DL – the Knowledge Management information space. Four users took part in the evaluation study.

To identify the relevant subset of documents for generating personalized extensions, two alternative approaches appeared feasable: *direct* monitoring of individual users web usage data, (i.e. information about access to websites and the like) or *indirect* usage profiling by means of explicit user profiles.

Direct Usage Context. In this approach, all information about access of the individual users to web sites is monitored explicitly or recorded by means of web browser history files which are then processed further, to obtain a usage context. From the access information, a set of recently visited documents of the BT DL is extracted. This set of documents was used as the input of the learning process. As a baseline for the evaluation, unpersonalized ontologies are generated from a corpus of a fixed number of random abstracts from the knowledge management information space of the BT Digital Library.

While this approach is feasable and easy to implement it appears unsuited for evaluation within this deliverable. On the one hand, experience shows that the usage record is typically very sparse leading to small document sets. On the other hand, to allow for a fair evaluation between different users, we would need to collect this usage information under experimental settings. As an alternative way of obtaining usage contexts, we considered the indirect usage context.

Indirect Usage Context. In this approach, the set of documents for learning personalized ontologies for a specific user is extracted by selecting those documents which were classified against at least of the topics of the set of topics that the user in question has marked as relevant (user profile).

Using the second approach, we are able to generate a much larger and far more stable selection of documents from the BT digital library for each user under identical settings for all users.

In our experiments, we used an ontology that was learned from abstracts of the KM information space without any user profile information as a baseline for the comparison. The personalized ontologies were obtained in the following way: We identified a relevant subset of documents from the information space based on the individual user profiles describing the interests. For this evaluation the profiles were created manually by the users indicating a set of relevant topics from the existing topic hierarchy [1]. The user-specific set of documents then was selected as the subset of documents that were classified

[1] In the final Digital Library these profiles would be created automatically, as described in [MG04].

	unpersonalized			personalized		
User	rel. concepts	concepts	% relevance	rel. concepts	concepts	% relevance
User 1	21	406	5,2%	8	135	5,9%
User 2	36	406	8,9%	55	566	9,7%
User 3	36	406	8,9%	18	139	12,9%
User 4	52	406	12,8%	25	141	17,7%

Table 3.1: Relevance of Concepts for the Personalized and Unpersonalized Ontologies

against at least of the topics of the user profile. Based on this data set, a personalized ontology for each user was learned using the same set of algorithms as for the baseline ontology. The personalized and unpersonalized ontologies were then presented to the users for evaluation who had to rate the usefulness of the learned ontology in terms of relevance of the learned concepts. To prevent a bias in the evaluation, the users were not informed beforehand that two different ontologies were to be evaluated, instead they were presented with the union of the concepts from both ontologies.

Evaluation Measures As the main goal of learning personalized ontologies is to generate structures that are relevant to the individual interests of the user, we considered as main evaluation measure the relevance of the identified concepts. Other important aspects for the evaluation of ontology learning are measures such as correctness in terms of precision and recall of identified elements, however, these are (1) typically not affected by individual preferences, (2) evaluated as part of deliverable [VS05b].

Further, we focused on the evaluation of the relevance of concepts, as they can be considered as “the backbone” of the ontology. The users simply had to mark the learned concepts as either relevant or irrelevant for their domain of interest. The relative relevance was then calculated as follows:

$$relevance = \frac{number\ of\ relevant\ concepts}{total\ number\ of\ concepts}$$

Results Table 3.1 shows the absolute and relative figures for the relevant concepts for the case of the unpersonalized and personalized ontology. The relevance is also shown as a diagram in Figure 3.1. A first observation is that the relative relevance is generally very low (between 5 and 18%). In other words, the results of current ontology learning techniques for identifying concepts to describe bibliographic entries are not satisfactory without further processing. One such processing step is already inherent in the learning process as changes to the ontology are not directly applied to the ontology, but presented to the user as *recommendations* for changes, which can be accepted or rejected. However, this also shows the need for techniques such as personalization of the learning process. We observe that a rather simple personalization based on the selection of documents according to the user profile results in a relative increase in relevance between 13% (User 1) and 45% (User 3). We believe that further personalization techniques e.g. based on feed-

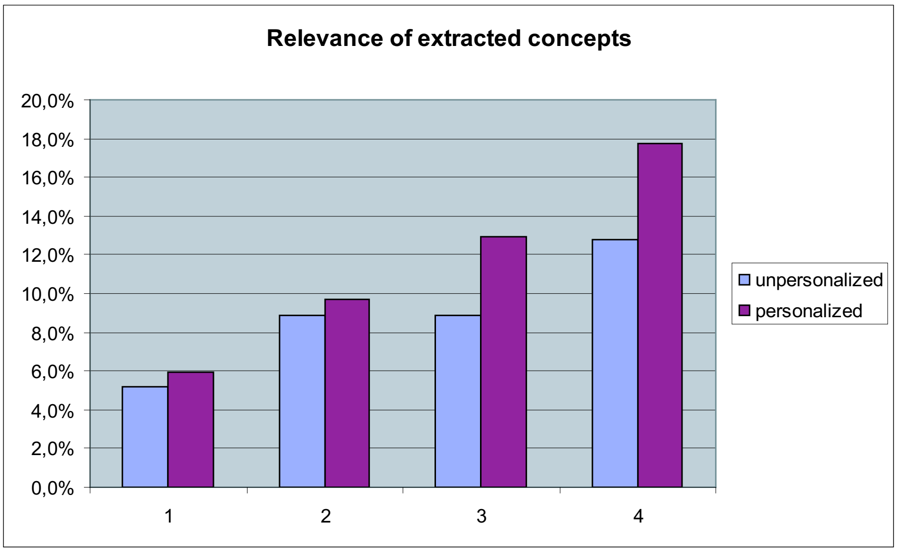

Figure 3.1: Relevance of Concepts for the Personalized and Unpersonalized Ontologies

back about previous accepted recommendation for ontology changes have the potential for further improvements in identifying relevant extensions to personalized ontologies.

Chapter 4

Conclusion

In this deliverable we have evaluated the framework of usage-driven change discovery by applying it to actual use cases from the BT DL case study. The use cases include (1) usage-based pruning of a generic ontology to obtain a personalized ontology, and (2) extending an existing ontology with personalized extensions obtained via Ontology Learning techniques. Further, these methods can be applied for generating information space specific ontologies based on usage data. The evaluation results show the usefulness of applying usage data for the task of ontology evolution.

The approach for usage-based pruning has been based on the idea of minimizing the cost to access documents in a repository with an ontology. Actual results are outstanding at this point in time, but will be delivered subsequently.

Generating personalized extensions on the other hand is based on the idea of adapting the ontology learning process by using user profile information obtained from usage information to identify a set of documents that will result in more relevant concepts in the personalized ontology. Evaluation results from an experiment with users of the BT DL show that such a personalization indeed improves the relevance of identified concepts considerably.

Bibliography

[BGM05] J. Brank, M. Grobelnik, and D. Mladenić. Ontology evaluation. SEKT deliverable 1.6.1, Jožef Stefan Institute, 2005.

[Dey01] A. K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.

[DM05] H. Dai and B. Mobasher. *Web Mining: Applications and Techniques*, chapter Integrating Semantic Knowledge with Web Usage Mining for Personalization. IRM Press, Idea Group Publishing, 2005.

[HS05a] P. Haase and Y. Sure. Ontology mangement and evolution - evaluation. SEKT deliverable 3.1.2, Institute AIFB, University of Karlsruhe, 2005.

[HS05b] P. Haase and Y. Sure. Usage tracking for ontology evolution. SEKT deliverable 3.2.1, Institute AIFB, University of Karlsruhe, 2005.

[MG04] Marko Grobelnik Miha Grcar, Dunja Mladenic. User profile specification and data gathering module. SEKT deliverable 5.5.1, (Jozef Stefan Institute), 2004.

[SMB04] A. Sieg, B. Mobasher, and R. Burke. Inferring user's information context: Integrating user profiles and concept hierarchies. In *Proceedings of the 2004 Meeting of the International Federation of Classification Societies, Chicago, IL, July 2004*, 2004.

[Sto04] L. Stojanovic. *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe, 2004.

[VS05a] J. Voelker and Y. Sure. Data-driven change discovery. SEKT deliverable 3.3.1, Institute AIFB, University of Karlsruhe, 2005.

[VS05b] J. Voelker and Y. Sure. Data-driven change discovery. evaluation. SEKT deliverable 3.3.2, Institute AIFB, University of Karlsruhe, 2005.