Ontology-Based Discovery of Geographic Information Services – An Application in Disaster Management

Eva Klien, Michael Lutz & Werner Kuhn
Institut für Geoinformatik
Robert-Koch-Str. 26-28, 48149 Münster
{klien|m.lutz|kuhn}@uni-muenster.de

Abstract. Finding suitable information in the open and distributed environment of current geographic information web services is a crucial task. Service brokers (or catalogue services) provide searchable repositories of service descriptions but the mechanisms to support the task of service discovery are still insufficient. One of the main challenges is to overcome semantic heterogeneity caused by synonyms and homonyms during keyword-based search in catalogues. This paper presents a practical case study to what extent ontology-based service discovery can solve these semantic heterogeneity problems. To this end, we apply the Bremen University Semantic Translator for Enhanced Retrieval as a service broker. The approach combines ontology-based metadata with an ontology-based search. Based on a scenario of finding geographic information services for estimating potential storm damage in forests, it is shown that through terminological reasoning the request finds an appropriate match in a service on storm hazard classes. However, the approach reveals some limitations in the context of geographic web service discovery, which are discussed at the end.

1 Introduction

Geographic information is the key to effective planning and decision-making in a variety of application domains. So-called intelligent web services permit easy access and effective exploitation of distributed geographic information (GI) for all citizens, professionals, and decision-makers (Bishr & Radwan, 2000; Brox, Bishr, Kuhn, Senkler, & Zens, 2002).

This paper focuses on the task of service discovery, which is a crucial task in the open and distributed environment of GI web services. Often, effective service discovery requires an extensive search for appropriate services across multiple application domains. Catalogues support discovery, organisation, and access of geographic information and thus help the user to find information that exists (OGC, 2004). However, the use of different vocabulary in the different application domains might lead to semantic heterogeneity problems when only simple keyword-based search is employed to find relevant information in a catalogue. Such problems arise when terms are unknown, the meaning of elements is not intuitively clear, or the understanding of the information provider differs from that of the requestor (Schuster & Stuckenschmidt, 2001).

Explication of knowledge by means of ontologies is a possible approach to overcome the problem of semantic heterogeneity, as ontologies can be used for the identification and association of semantically corresponding concepts (Wache, Vögele, Visser, Stuckenschmidt, Schuster, Neumann et al., 2001). The task of ontology-based information brokering has been addressed by the Bremen University Semantic Translator for Enhanced Retrieval (BUSTER) (http://www.semantic-translation.de/). Among other things, this system provides an ontology-based approach with logical reasoning on metadata for retrieving information sources (Neumann, Schuster, Stuckenschmidt, Visser, & Vögele, 2001).

The ideas presented in this paper are well known and applied in the context of the Semantic Web (http://semanticweb.org/). To increase precision and recall during service discovery in current service
registries (e.g. OGC catalogue services or UDDI registries), several approaches that are based on reasoning with semantic service descriptions that refer to ontologies have been proposed (Kawamura, De Blasio, Hasegawa, Paolucci, & Sycara, 2003; Paolucci, Kawamura, Payne, & Sycara, 2002; Sirin, Hendler, & Parsia, 2003). Several XML-based mark-up languages are available for the description of ontologies including RDF Schema (W3C, 2004) and OWL (Antoniou & Van Harmelen, 2003).

We have decided to focus our work on the mechanisms of semantic matchmaking by means of terminological reasoning. In order to remain independent from current web implementations, we use a basic description logic as a representation language. The presented example can easily be reconstructed using a terminological reasoner like RACER. We examine to what extent this approach can contribute to solving semantic heterogeneity problems that can occur during GI service discovery. To this end, we apply BUSTER as service broker in the GI service discovery scenario of this work. Our long-term goal is to integrate the mechanisms presented in this work into geographic services architectures (Klien, Einspanier, Lutz, & Hübner, 2004).

The remainder of this paper is structured as follows. In section 2 we describe a motivating example for our research. Section 3 gives an introduction to GI web service discovery and points out semantic heterogeneity problems that may arise. The approach that is applied for ontology-based GI web service discovery is presented in section 4, followed by the method descriptions in section 5. In section 6 the capabilities of the approach are examined by applying the presented approach for service discovery to the motivating example. The paper concludes with a discussion of the problems encountered and an outlook on the remaining and emerging research questions.

2 Motivating Example: Discovering Services for Estimating Storm Damage in Forests

The motivating example is set in the area of disaster management and mitigation. Heavy storms, such as the winter storm “Lothar” over Central Europe in December 1999, may cause severe road blockage by windfall timber. We use the motivating example throughout the paper to illustrate problems caused by semantic heterogeneity and to apply the presented approach for service discovery. However, our work is designed to be domain independent and is not restricted to only this example.

Susan is the official on duty in the regional authority responsible for ensuring road safety. After a heavy storm, she coordinates the assignment of the Governmental Disaster Relief Organisation (Technisches Hilfswerk, THW) and of the Federal Armed Forces (Bundeswehr) in the affected areas on a multi-regional scale. Susan has to keep track of where and to what extent the local authorities need help in order to clear the road blockages as quickly as possible.

In order to coordinate the clearing operations effectively Susan requires an overview of which roads are most likely to be affected by fallen timber. She can obtain this overview by overlaying road data with information on potential storm damage in the forests of the region. In order to do that she first has to acquire information of the susceptibility of forests to storm damage. Finding suitable information sources is the focus of this scenario.

The availability of geographic information with adequate quality (currency, completeness) is crucial for sound decisions at a local, regional and global level. GI web services are a key technology in compiling and providing the necessary information for decision makers in an ad hoc fashion (Bernard, Einspanier, Haubrock et al., 2003). Since each situation will require different information to solve the problem at hand, service discovery, i.e. finding suitable services for answering a given question, becomes a crucial task.

3 GI Web Service Discovery

GI web services offer information products rather than raw datasets only interpretable by geographic information system (GIS) experts. The open and distributed GI web service environment opens a wide
range of new possibilities for acquiring, processing and analysing geographic information without the need of GIS expert knowledge (Greve, 2002).

The World Wide Web supplies the basic infrastructure for system interoperability, i.e. distributed use and multiple exploitation of data and systems. Furthermore, the Open Geospatial Consortium (OGC) and the International Organisation for Standardisation (ISO) have developed geoinformation technology standards providing the essential basis for syntactic interoperability and cataloguing of GI web services (Bernard, Einspanier, Haubrock et al., 2003).

In an environment where services are previously unknown, a service that is appropriate for answering a given question from among a large number of available services has to be discovered first. Service discovery, thus, is a crucial task that will become even more important with the emerging Semantic Geospatial Web (Egenhofer, 2002).

3.1 General (GI) Web Service Architecture

Figure 1 depicts the general “publish-find-bind” pattern of web service architectures, which has been adopted by OGC and ISO (ISO/TC-211 & OGC, 2002). Three roles can be identified. Service providers offer data or applications as services. A service is published with a service broker by advertising a declarative metadata description of the service’s properties, e.g. its input, output or performance. Service requestors search for services that provide the information needed to solve the problem at hand (Nebert & Reed, 2002). The find-operation thus involves searching for an appropriate service by querying the service broker for relevant matches for a given question. A Service broker (or catalogue service) is an intermediary service whose responsibility is to bring a service requester and a service provider together (OGC, 1996), and thus may be considered the core of any GI web service environment.

Figure 1: The „publish, find and bind“ pattern of web service architectures ((Zhang & Hutchison, 2002), modified).

After a service offered by a provider is identified to match the service requestor’s requirements it is bound to the service requestor, the service then is executed, passing data and instructions across common interfaces (Nebert & Reed, 2002).

3.2 Problems caused by Semantic Heterogeneity during Service Discovery

Although standards from bodies like the OGC provide the basis for syntactic interoperability the usability of information that is created in one context is often of limited use in another context (Bernard, Einspanier, Haubrock et al., 2003), because of insufficient means for meaningful interpretation. This problem is referred to as the need for “semantic interoperability among autonomous and heterogeneous systems” (Goh, Bressan, Madnick, & Siegel, 1999). Problems caused by semantic heterogeneous descriptions play a crucial role during the task of finding relevant information within a GI web service environment (Lutz, Riedemann, & Probst, 2003).

Heterogeneity is an inherent problem in the geoscientific area because of the wide variety of potential applications. In Bishr (1998), semantic heterogeneity is defined as the consequence of different con-
ceptualisations and database representations of a real world fact. Two types of semantic heterogeneity can be distinguished:

- **Cognitive heterogeneity**: Because of different perspectives on the same real world facts there may not be a common base of definitions of the underlying facts between two disciplines (domains). Problems can occur if these cognitive differences are concealed because the same term is used for different concepts.

- **Naming heterogeneity**: The same real world facts are understood in the same way but are named differently.

The problems that semantic heterogeneity can cause during GI web service discovery can be illustrated by extending our motivating example introduced in section 2. John is a forest ecologist who has developed a model for calculating storm hazard classes for forest stands on the basis of five influencing factors. In order to make his results open to public, John publishes his model as a GI web service. This service returns the forest stands (polygons) of a specified area classified in storm hazard classes. John has developed his model independently from Susan’s question, but nevertheless, the resulting information may be used for estimating damage after storms (as well as for a variety of other applications, e.g. sustainable forest management).

It is not necessarily deducible for a non-expert in forest ecology that the service calculating storm hazard classes could also be used for estimating storm damage. Both types of semantic heterogeneity introduced in Bishr (1998) can lead to problems if Susan performs a simple keyword-based search, e.g. using the terms “estimate storm damage”:

1. **Naming heterogeneity**: If John has described his model in its specific context with the terms “storm hazard model” Susan will fail to find the storm hazard service although it offers relevant answers for her question (figure 2).

2. **Cognitive heterogeneity**: Susan’s keyword search could also result in finding services that are not appropriate for answering her question, thus indicating the occurrence of cognitive heterogeneity. This would be the case if, for example, a service for depicting storm damages in forests over the last three decades was annotated with keywords like “storm damage” and “forest”.

![Figure 2: Semantic heterogeneity problems caused by different domain perspectives and terminologies.](image)

These examples show that keywords used in free-text entries have to be considered a poor way to capture the semantics of a query or item (Bernstein & Klein, 2002).

Consequently, in order to solve these heterogeneity problems an approach is needed that exceeds the capabilities of current keyword-based search facilities in catalogues. So far, the problem has been acknowledged (Bernard, Einspanier, Lutz, & Portele, 2003; Bishr & Radwan, 2000; Egenhofer, 2002; Lutz, Riedemann, & Probst, 2003) but still no standardised technological solution exists within the GI web service community.
Accepting the diversity of geographic application domains, such an approach would need to enable navigating differences in meaning (Harvey, Kuhn, Pundt, Bishr, & Riedemann, 1999). In Stuckenschmidt (2002), it is suggested that explicit context models be used to re-interpret information in the context of a new application. Ontologies have become popular in information science as they can be used to explicate contextual information. In the following we present an ontology-based approach for overcoming semantic heterogeneity problems during service discovery, which has been realized in the BUSTER system.

4 Ontology-Based Approach to Service Discovery

BUSTER is a scientific prototype developed at the Centre for Computing Technologies (TZI) in Bremen for supporting the specific tasks of information retrieval and information integration in distributed and heterogeneous environments. The BUSTER system offers functionalities in order to solve heterogeneity problems on three different levels (syntactic, schematic and semantic) by combining several technologies including standard markup languages, mediator systems, ontologies and knowledge based classifiers (Visser, Stuckenschmidt, Wache, & Vögele, 2000). In the context of GI web service discovery BUSTER becomes interesting as one of the functionalities it provides is an ontology-based search with terminological reasoning on metadata for finding information sources (Neumann, Schuster, Stuckenschmidt, Visser, & Vögele, 2001).

In this section we present the general approach employed in BUSTER for ontology-based search for information sources. How we apply this approach to GI web services is described in section 5.

4.1 Ontologies

The term “ontology” has been used in information sciences with several meanings. Gruber (1993) introduced the term “ontology” to mean an “explicit specification of a conceptualization”. This initial definition was slightly modified in Borst (1997), where ontologies are defined as “a formal specification of a shared conceptualization”. Merging both definitions, we define an ontology as “an explicit formal specification of a shared conceptualization” (Studer, Benjamins, & Fensel, 1998). According to Uschold (1998) a conceptualisation is the way of thinking about some domain. A conceptualisation may be implicit, e.g. existing only in one’s head, or embodied in a piece of software. To make it explicit means to define both the type of concepts used and the constraints on their use (Benjamins, Fensel, & Gomez-Perez, 1998). Finally, in order to make it machine-readable, it has to be formalised in some representation language.

All this adds up to making the ontology a perfect candidate for communicating a shared and common understanding of some domain across people and computers (Studer, Benjamins, & Fensel, 1998).

4.2 Hybrid Ontology Approach

The ontology approach of BUSTER is based on the idea of having a source-independent shared vocabulary for one domain (figure 3).
It is assumed that the members of a domain share a common understanding of certain concepts, i.e. no further explication is needed. These concepts form the basic terms contained in a shared vocabulary. The shared vocabulary is usually built by an independent domain expert, who is familiar with typical tasks and problems in a domain.

Once a shared vocabulary exists, the terms can be used to make explicit the contextual information of the information sources that are to be integrated (Visser & Stuckenschmidt, 2002), e.g. to build an application ontology for an information source. Thus the vocabulary has to be general enough to be used across all information sources that are to be annotated within the domain, but specific enough to make meaningful definitions possible (Schuster & Stuckenschmidt, 2001). The task of constructing an application ontology lies in the responsibility of the provider of the information source.

4.3 Ontology-Based Metadata

In order to register an information source with BUSTER, a metadata description in form of a Comprehensive Source Description (CSD) is needed. Each CSD consists of metadata that describe technical and administrative details of the data source as well as its structural and syntactic schema and annotations (Visser, Stuckenschmidt, Wache, & Vögele, 2000).

The CSD is based on the metadata standard Dublin Core and formalized in XML/RDF. Visser & Stuckenschmidt (2002) state that using XML (Extensible Markup Language) is a suitable way of exchanging data with a well defined syntax and structure and that simple RDF (Resource Description Framework) provides a uniform syntax for exchanging meta-information in a machine-readable format. The application ontology is referenced in the CSD and thus adds capabilities for reasoning about meaning.

Setting up a CSD is the task of the provider of the information source.

4.4 Knowledge Representation Language

A knowledge representation language is used to formally describe the concepts of an ontology. In order to perform the tasks of searching, of discovering relationships among concepts, and of looking for inconsistencies in the ontology, the meaning of concepts in the ontology must be represented in a way that can be manipulated by machines. There are a variety of languages that can be used for the representation with varying characteristics in terms of their expressiveness, ease of use and computational complexity (Stevens, Goble, & Bechhofer, 2000).

Description Logics (DL) is a family of languages that describe knowledge in terms of concepts and restrictions on roles. The main idea behind DL is to provide means to describe structured knowledge in a way that can be accessed and reasoned with (Nebel, 1996). DL theory is divided into a termino-
logical part (TBox) and an assertional part (ABox). TBox deals with the definition of concepts, while ABox asserts facts about individuals (single objects).

With respect to the applied reasoner in the BUSTER system (see next paragraph), the DL language used for representing the concepts of the shared vocabulary and the application ontology is SHIQ. The “S” stands for the basic DL that SHIQ is based on, i.e. ALC R+. This basic DL is extended with role hierarchies (“H”), inverse roles (“I”), and qualifying number restrictions (“Q”) (Horrocks, Sattler, & Tobies, 2000).

Generally, SHIQ enables algorithms for TBox reasoning as well as for ABox reasoning. However, in the current prototype of the BUSTER system only TBox reasoning is considered, i.e. the shared vocabulary and the application ontologies of a domain are implemented in the same TBox. For more information on SHIQ see Horrocks, Sattler, & Tobies (2000).

4.5 Ontology-Based Search

BUSTER, the task of automatically mapping between concepts of different application ontologies within the same domain is performed by the DL system RACER (Reasoner for A-Boxes and Concept Expressions Renamed). The process is called semantic translation through re-classification (Visser, Stuckenschmidt, Wache, & Vögele, 2000), i.e. BUSTER allows the classification of data into another context through subsumption reasoning.

Determining whether one description subsumes another one, that is, whether the first is more general than the second is one important reasoning task of DL systems. Formally, subsumption can be defined as follows: In a terminology T containing concepts C and D, C is subsumed by D if in every model of T the set denoted by C is a subset of the set denoted by D (Donini, 2003).

With subsumption tests, one can organize the concepts of a terminology into a hierarchy according to their generality. A concept description can also be conceived as a query, describing a set of objects one is interested in Donini (2003). Thus, all concepts that are subsumed by the query concept can be considered to also satisfy the query.

A number of algorithms exist to compute subsumption. In order to present the general idea, we introduce a structural subsumption algorithm for a simple DL (FL0) taken from Donini (2003)\(^1\):

\[ A_1 \land \ldots \land A_m \land \forall R_1.C_1 \land \ldots \land \forall R_n.C_n \]

be the normal form of the FL0-concept description C, and

\[ B_1 \land \ldots \land B_k \land \forall S_1.D_1 \land \ldots \land \forall S_n.D_n \]

be the normal form of the FL0-concept description D. Then C subsumes D iff the following two conditions hold:

(i) for all i; 1 \leq i \leq k, there exists j; 1 \leq j \leq m such that \( B_i = A_j \).

(ii) For all i; 1 \leq i \leq l, there exists j; 1 \leq j \leq n such that \( S_i = R_j \) and \( C_j \) subsumes \( D_i \).

This proposition can be illustrated in the following (simplified) example: The concept broadleafForest is defined as a subconcept of landuseCategory whose mainVegetationType role is restricted to broadleafTree. Now it is possible to define a queryConcept as a subconcept of landuseCategory whose mainVegetationType role is restricted to Tree\(^2\). As both broadleafForest and queryConcept have the same superconcept, and Tree subsumes broadleafTree, it can be deduced that queryConcept subsumes broadleafForest, and that hence this concept represents an answer to the query represented by the queryConcept.

\(^1\) For a description of subsumption algorithms for more expressive DLs, see Donini (2003)

\(^2\) Of course, this concept could also be called forest.
5 Applying the BUSTER Approach to GI Service Discovery

In the following, the methods for applying the BUSTER approach to the GI web service discovery scenario are presented.

5.1 BUSTER as Service Broker

In general, ontology-based search with BUSTER can be divided in two phases: a data acquisition phase and a query phase. The first phase includes all necessary server-side preparations for registering an information source with BUSTER. The second phase refers to the application of the BUSTER client by a human user in order to find a specific information source.

When applying the BUSTER approach to the “publish-find-bind” pattern of web service architectures the acquisition phase corresponds to the publish-operation (figure 4). John, the service provider in our motivating example, can publish his storm hazard service by registering the BUSTER-specific CSD for the service. The registration of a CSD is essential to make the service detectable via the BUSTER client application.

The service requestor (i.e. Susan) searches for appropriate services by defining a query for a service for estimating storm damage in forests during the query phase of BUSTER (which corresponds to the find-operation).

![Diagram of service discovery scenario](image)

**Figure 4:** The service discovery scenario transferred to the “publish-find-bind” pattern.

As described in the previous section the following tasks have to be accomplished in order to prepare and to conduct the publish-operation for the storm hazard service within the BUSTER framework:

1. A **shared vocabulary** for (at least for parts of) the domain of forest ecology has to be defined.
2. An **application ontology** for the service’s output (i.e. storm hazard classes) based on the shared vocabulary has to be built.
3. Finally a **CSD** for the service on storm hazard has to be written.

5.2 Defining a Shared Vocabulary

For building a shared vocabulary in the domain of forest ecology, the method introduced in Schuster & Stuckenschmidt (2001) is adopted. The authors suggest an iterative approach through five steps:

1. **Finding bridge concepts:** Bridge concepts are query templates that contain admissible combinations of properties and property values, which can be seen as “points of entry” into the
shared vocabulary of a domain. After choosing a query template, users can use the provided properties and property values to specify value constraints for their query. This procedure significantly reduces the amount of terms that are presented to the user and prevent inexperienced users from defining queries that do not make sense.

2. **Defining properties**: The builder of the shared vocabulary has to define properties that describe the chosen bridge concept.

3. **Finding property values**: The property values are the “fillers” of the defined properties. They are the main part of the shared vocabulary since they are used to define the concepts in the application ontology, which is built later on.

4. **Adapting the shared vocabulary**: During the first development cycles, the shared vocabulary will probably not be expressive enough. (Schuster & Stuckenschmidt, 2001) suggest to adapt the shared vocabulary by building a special "support ontology" in order to revise the problem.

5. **Refining the definitions**: Following the "evolving” life cycle, the engineer can step back all the time to modify, add and remove ontology definitions.

For identifying bridge concepts, properties and property values, several sources of information have been chosen. Two standard works on forest ecology are examined: Barnes, Zak, Denton, & Spurr (1998) and Otto (1994). Furthermore the “General Multilingual Environmental Thesaurus” (GEMET) is taken into account. GEMET provides a core terminology of generalized environmental terms and definitions, covering more than 5,400 terms (Nax & Lethen, 1999). Within the scope of this work, the examination focuses on terms dealing with disturbances and the susceptibility in an ecosystem.

Furthermore, the authors decided to extend this approach by taking into account the specific requirements of describing a *service* rather than an *information item*. It is an unrealistic expectation to find a complete set of characteristic attributes for a real world object. Describing objects based on their use, i.e. based on their function in the cognitive world, however, has the potential to overcome this problem (Bishr, 1998). Correspondingly Kuhn (2001) suggests, that in order to make geographic information more useful and usable, ontologies should be designed with a focus on human activities in geographic space. This approach seems reasonable for defining a shared vocabulary that is to be integrated into a broker for web services. GI web services are needed in order to answer specific questions, i.e. a service requestor recognises objects based on their use. Therefore, the shared vocabulary should provide respective concepts that indicate functions in terms of “is useable for”. For that reason, in addition to the concepts identified by the method introduced above, the actions or tasks (i.e. functions) related to a bridge concept are included as property values.

5.3 **Building an Application Ontology**

The application ontology is the formalised, explicit description of the GI service for modelling storm hazard. In the context of this work, only the information output of this service is relevant. For describing the respective concept a clear understanding of the underlying model is needed.

John’s model of storm hazard calculates the degree of the susceptibility of a forest stand to storm. The result is expressed in a *storm hazard class*. The impact of a storm depends on the susceptibility of a forest stand at that particular moment. Therefore each storm hazard class (indicating the susceptibility of a forest stand) has assigned a critical wind speed (Table 1).

<table>
<thead>
<tr>
<th>Storm hazard class</th>
<th>Critical wind speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>A very high</td>
<td>&gt; 60 km/h</td>
</tr>
<tr>
<td>B high</td>
<td>&gt; 80 km/h</td>
</tr>
<tr>
<td>C intermediate/average</td>
<td>&gt; 120 km/h</td>
</tr>
<tr>
<td>D inferior</td>
<td>&gt; 200 km/h</td>
</tr>
</tbody>
</table>
The model can be useful in a variety of applications. It can be applied for sustainable forest management, predictions on occurrences of damage, and advanced warnings for all kinds of stakeholders related to forests. With the additional knowledge about the maximum wind velocity over a forest stand, a rough estimation of damage after a storm is possible as well.

John has to describe the output of his service (i.e. the concept of a storm hazard class) by using only terms offered in the shared vocabulary of the forest ecology domain. On this basis, he tries to make the semantics of a storm hazard class as explicit as possible.

5.4 Writing a CSD

Writing a CSD does not require a specific method. The structure is specified by the requirements of BUSTER (section 4.3) and can be adopted for any new service to be published.

While it is possible to include references to multiple concepts of the application ontology in each CSD, it cannot be explicitly stated which of the service properties (e.g. input, output, functions) the concept refers to. In order to annotate all properties of a service a modification of the CSD or of the reasoning process would be required. This is further discussed in section 7.

In our example, Susan is interested in finding information for estimating storm damage in forests, i.e. the information output of a service. Therefore to publish a CSD that includes a reference only to the concept of the service’s output is sufficient for making the find-operation feasible.

6 Publishing and Discovering a GI Service with the BUSTER Approach

This section describes how the presented approach for ontology-based service discovery can be applied to our motivating example. We first present the shared vocabulary, the application ontology and the CSD required for registering the service in BUSTER (sections 6.1 to 6.3)\(^3\). Secondly, we apply these components in order to realise Susan’s query for services for estimating storm damage in forests (section 6.4).

6.1 The Shared Vocabulary

There are a variety of bridge concepts that could be defined for the domain of forest ecology. In the context of our scenario the bridge concept disturbance is relevant. The properties and property values in Table 2 are only a small extract from the totality of possible values, but they are sufficient for the purpose of this work. As introduced in section 5.2, the notion of functions (is usable for) is also considered. The shared vocabulary is registered with BUSTER.

Table 2: Extract of the shared vocabulary for the domain of forest ecology

<table>
<thead>
<tr>
<th>Bridge Concept</th>
<th>Properties</th>
<th>Range</th>
<th>Fillers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disturbance_Measure</td>
<td>_disturbance_caused_by</td>
<td>Cause</td>
<td>Storm, Fire, Acidification, Pest_Infestation</td>
</tr>
<tr>
<td></td>
<td>_disturbance_affects</td>
<td>Affected_landuse</td>
<td>Vegetation, Fauna, Soil</td>
</tr>
<tr>
<td></td>
<td>_disturbance_occurs_in</td>
<td>Affected_entity</td>
<td>Forest, Pasture, Wetland</td>
</tr>
</tbody>
</table>

\(^3\) The ontologies and CSD used for this application are stored and accessible on the following website: http://ifgi.uni-muenster.de/~klien/sources/
6.2 The Application Ontology

Figure 5 contains an extract of the application ontology, describing the general concept of a storm hazard class (section 5.3) by using the shared vocabulary of the forest ecology domain.

\[
\begin{align*}
&\text{(define-concept Storm-Hazard-Class} \\
&(\text{and} \\
&(\text{some_disturbance_caused_by Storm}) & a \\
&(\text{some_disturbance_affects Vegetation}) & b \\
&(\text{some_disturbance_occurs_in Forest}) & c \\
&(\text{some_disturbance_occurs_in Pasture}) & e \\
&(\text{some_is_produced_by Simulation}) & e \\
&(\text{some_is_usable_for Depiction}) & e \\
&(\text{some_is_usable_for Regeneration}) & f \\
&(\text{some_is_usable_for Estimating_Damage}) & f \\
&(\text{some_critical_wind_speed WindSpeed}) & g \\
&\text{)} \\
\end{align*}
\]

**Figure 5:** Notation for the concept of a storm hazard class.

In the following, the single expressions used for describing the concepts are explained. A storm hazard class describes a disturbance, which is caused by a storm (a), affects vegetation (b) and occurs in forests or pastures (c). It is produced by a simulation (e), can be used for depicting or regenerating a disturbance or for estimating the damage caused by it (f), and has a specific critical windspeed (g).

The application ontology comprises more concepts, e.g. model of storm hazard which is defined via the input influencing factors and the output concept storm hazard class. However, as for our example only the semantics of the service’s information output are of interest, these other concepts are not further described here.

The application ontology can be put on any server for access. The web address is referenced in the CSD of the GI web service.

6.3 The Comprehensive Source Description

The CSD of the service for modelling storm hazard comprises a reference to the concept storm hazard class in the <subject> tag. The respective application ontology is referenced in the <reference> tag (figure 6). The other tags contain administrative and technical details that become interesting once a service has been identified as being suitable for matching a specific query.
The CSD is stored on any server and registered via its web address with BUSTER.

6.4 Query Phase

Once the service for modelling storm hazard has been annotated with all the information needed, it can be found via the client application of BUSTER. The system offers two different types of queries: The simple query, which uses existing concepts from application ontologies, is not further discussed in this paper. The other type is the defined concept query, which allows the user to define a concept based on a given shared vocabulary, which fits his understanding of a concrete concept (Visser & Stucken-schmidt, 2002). In the following, this user defined concept is referred to as query concept.

Figure 7 shows the procedure of the defined concept query in the context of our scenario, i.e. the query for services for estimating storm damage in forests.

In our motivating example, after having chosen the domain of forest ecology, Susan selects a query-template provided by the BUSTER server, which is based on the domain’s shared vocabulary. The
templates correspond to the bridge concepts of the shared vocabulary and contain its properties (slots) and values (filler). This ensures, that Susan only makes use of well-known terms when defining a query concept.

Figure 8 shows the selection options offered by the query-template Disturbance_Measure. On this basis Susan is asked to select reasonable values for the given properties.

Susan chooses the fillers STORM from the property DISTURBANCE_CAUSED_BY and the filler ESTIMATING_DAMAGE from the property IS_USABLE_FOR. She is not interested in services that deal with the other options. The filled query-template is then automatically translated into a logical term. Susan is not concerned with the task of formalising the explicit description of the query concept.

During the query process all CSDs related to the current domain of forest ecology are parsed for the <subject> tag. The <reference> tag points to the respective application ontology (see figure 6). These ontologies are downloaded and transferred into the RACER inference machine. After reclassification (section 4.5), all sub-concepts of the query concept are presented as results (Visser & Stuckenschmidt, 2002).

Susan’s query results in the discovery of the service on storm hazard, as the concept of storm hazard class is identified as being a subconcept of the user’s query concept. Figure 9 depicts the results in RACER (visualised with the graphical user interface RICE). The relevant concepts of the shared vocabulary are shown on the left. The result of the reclassification of the concept of storm hazard class in the new context of the query concept is shown on the right.

---

4 You can try this example at http://geoshare.tzi.de/ConceptDefinitionClient/. The application is under constant development and therefore subject to change. This may affect the appearance or feasibility of this example.
Apart from the scenario query, there are a variety of other defined concept queries that would succeed in discovering the service on storm hazard, e.g. searching for information in order to depict disturbances that affect vegetation.

7 Conclusions and Future Work

The BUSTER approach uses ontology-based metadata in combination with terminological reasoning to ensure semantic interoperability during information retrieval. The approach has been developed for well-defined domains in which a small number of people or institutions commit to a common shared vocabulary. In this paper, we apply the approach in the broader context of GI web service discovery. The defined concept query offered by BUSTER has proven capable of solving semantic heterogeneity problems caused by naming heterogeneity and cognitive heterogeneity (section 3.2):

- The definition of the query concept is based on the shared vocabularies that are registered for different domains, i.e. the use of valid terms is ensured. In this manner, the problems that may arise due to naming heterogeneity during simple keyword-based search are avoided.
- The terminological reasoning facility of BUSTER enables concept (rather than only syntactic) matching and ensures that problems caused by cognitive heterogeneity are factored out as well.

While these results indicate that the approach is promising in the chosen application context, a number of challenges remain.

**Publishing a service.** BUSTER has been designed to deal with single information items. In such cases, the reference to one concept description is sufficient during publishing. Publishing GI web services, however, is more complex, as a service is characterised through several properties, e.g. input, output, functionality and performance (Bernstein & Klein, 2002). The method for constructing an application ontology for web services should consider these requirements, i.e. the concept definitions should reflect the service properties that will be searched for later. It is to be examined to what extent the shared vocabulary, the application ontology, and the CSD can meet these additional requirements for publishing GI web services.

While in the presented study, only a concept for the service’s output information is referenced in the CSD, the CSD could in principle comprise references to more than just one concept of an application ontology. Thus, it fulfils an important requirement for annotating web services, i.e. each service property can be annotated with the corresponding concept in the application ontology.

**Creating shared vocabularies.** A shared vocabulary provides a common basis for the interpretation of the application ontologies built using the vocabulary’s terms. Thus, a shared vocabulary ensures semantic interoperability between all organisations committing to it. However, this central role also comes at a price: As the a shared vocabulary claims to comprise the basic terms of a common conceptualisation of a domain, great care must be taken to define the terminology on an appropriate level of expressiveness. The terms have to be general enough to allow the annotation of all information sources, but specific enough to make meaningful definitions possible. In consequence, a shared vocabulary will only be useful if it is defined within a certain context and for a well-known user community. The definition of sound shared vocabularies becomes ever more complicated in case of GI web services.
service environments, which comprise a variety of application domains and whose boundaries are hard
to define. For example, consider the domain of forest ecology in relation to soil ecology, vegetation
ecology, landscape ecology, and similar disciplines.

Furthermore, the restriction of shared vocabularies to a few bridge concepts, their properties and al-
lowed property values leads to a severely limited expressiveness. For example, the shared vocabulary
in our example would be more intuitively modelled using two concepts representing the disturbance
and the disturbance measure, which are connected by the property DESCRIBES (Table 3). Unfortu-
nately, with the currently available prototype it is not possible to express an application ontology or
query based on such a shared vocabulary. A first approach to make shared vocabularies more expres-
sive and flexible is presented in Klien, Einspanier, Lutz, & Hübner (2004).

Table 3: Alternative shared vocabulary for the domain of forest ecology

<table>
<thead>
<tr>
<th>Bridge Concept</th>
<th>Properties</th>
<th>Property values</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISTURBANCE</td>
<td>CAUSED BY</td>
<td>see Table 2</td>
</tr>
<tr>
<td></td>
<td>AFFECTS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OCCURS_IN</td>
<td></td>
</tr>
<tr>
<td>DISTURBANCE MEASURE</td>
<td>DESCRIBES</td>
<td>DISTURBANCE</td>
</tr>
<tr>
<td></td>
<td>IS PRODUCED BY</td>
<td>see Table 2</td>
</tr>
<tr>
<td></td>
<td>IS_USABLE_FOR</td>
<td></td>
</tr>
</tbody>
</table>

Discovering a service. The presented approach for ontology-based search rests on the assumption that
all users have a common understanding of the terms provided in the shared vocabulary. This presents a
major difficulty if – as in the scenario presented in this work – service requestor and provider are from
different domains. In such cases, there has to be at least some overlap between the two actors’ concep-
tualisations (figure 10). However, this cannot always be assumed, especially if a user is not familiar
with the domain he requests information from. This inherent problem of using shared vocabularies
illustrates the need for grounding the used terms in conceptualisations. Approaches like the theory of
semantic reference systems introduced by Kuhn (2003) show promising steps in the right direction and
are subject of future research.

![Figure 10: Overlap of the conceptualisations of two different domains.](image)

Description Logic reasoning. Another drawback of the approach could be seen in the matchmaking
by DL reasoning, which is generally related to high computational complexity (Donini, 2003). In
BUSTER, the reasoning task has only been tested within small domains comprising only a few appli-
cation ontologies. Using BUSTER in an open GI web services environment with complex shared voc-
abularies and also more application ontologies to reason on, could make the task of matching con-
cepts during service discovery extremely slow. However, Li & Horrocks (2003) have shown that while
the average time (per registered data set) for classifying a TBox indeed increases rapidly with the size
of the registry, matchmaking in an already classified TBox is extremely fast. The authors therefore propose to classify the TBox offline before the matchmaking process starts, and use the classified TBox to reason about requests.

Integration into Spatial Data Infrastructures. Spatial Data Infrastructures are currently set up on local, regional, national and global level to facilitate the discovery and access to geographic data and GI services. In order to make the presented approach available to a wider community, it should be integrated with existing standardized SDI components such as Catalogue services (OGC, 2004). A first step in this direction is presented in Klien, Einspanier, Lutz, & Hübner (2004).

Future work will address the discussed problems and alternatives in order to further improve the BUSTER approach for GI service discovery. In addition, we will compare the overall approach to other algorithms for discovering web services, e.g. Sycara, Klusch, Widoff, & LU (1999).

Acknowledgements

The work presented in this paper has been supported by the German Federal Ministry for Education and Research as part of the GEOTECHNOLOGIEN program (grant number 03F0369A). It can be referenced as publication no. GEOTECH-[TBD]. We are grateful to Sebastian Hübner for his input at various stages of this work, especially for providing support on the BUSTER system and for enabling the implementation of the motivating example in the BUSTER prototype. The comments from four anonymous referees provided useful suggestions to improve the content of the paper.

References


