DOLCE ergo SUMO: On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO)

Daniel Oberle [a,h], Anupriya Ankolekar [a], Pascal Hitzler [a,*], Philipp Cimiano [a], Michael Sintek [b], Malte Kiesel [b], Babak Mougouie [b], Stephan Baumann [b], Shankar Vembu [h], Massimo Romanelli [c], Paul Buitelaar [c], Ralf Engel [c], Daniel Sonntag [c], Norbert Reithinger [c], Berenike Loos [d], Hans-Peter Zorn [d], Vanessa Micelli [d], Robert Porzel [e], Christian Schmidt [f], Moritz Weiten [f], Felix Burkhardt [g], Jianshen Zhou [g]

[a] *Institut AIFB, Universität Karlsruhe, Germany*
[b] *DFKI KM, Kaiserslautern, Germany*
[c] *DFKI IUI & LT, Saarbrücken, Germany*
[d] *European Media Lab (EML), Heidelberg, Germany*
[e] *Arbeitsgruppe Digitale Medien, Universität Bremen, Germany*
[f] *Ontoprise GmbH, Karlsruhe, Germany*
[g] *T-Systems, Berlin, Germany*
[h] *Now at SAP Research, CEC Karlsruhe, Germany*

Abstract

Increased availability of mobile computing, such as personal digital assistants (PDAs), creates the potential for constant and intelligent access to up-to-date, integrated and detailed information from the Web, regardless of one's actual geographical position. Intelligent question-answering requires the representation of knowledge from various domains, such as the navigational and discourse context of the user, potential user questions, the information provided by Web services and so on, for example in the form of ontologies. Within the context of the SmartWeb project, we have developed a number of domain-specific ontologies that are relevant for mobile and intelligent user interfaces to open-domain question-answering and information services on the Web. To integrate the various domain-specific ontologies, we have developed a foundational ontology, the SmartSUMO ontology, on the basis of the DOLCE and SUMO ontologies. This allows us to combine all the developed ontologies into a single SmartWeb Integrated Ontology (SWIntO) having a common *modeling basis* with *conceptual clarity* and the provision of *ontology design patterns* for modeling consistency. In this paper, we present SWIntO, describe the design choices we made in its construction, illustrate the use of the ontology through a number of applications, and discuss some of the lessons learned from our experiences.

Keywords: Ontology management; Foundational ontology; Core ontology; Domain ontology; DOLCE; SUMO; Mobile computing; Ambient intelligence; SmartWeb project

* Corresponding author. Tel.: +49 721 608 4751; fax: +49 721 608 6580.

E-mail addresses: d.oberle@sap.com (D. Oberle), ankolekar@aifb.uni-karlsruhe.de (A. Ankolekar), hitzler@aifb.uni-karlsruhe.de (P. Hitzler), cimiano@aifb.uni-karlsruhe.de (P. Cimiano), sintek@dfki.de (M. Sintek), kiesel@dfki.de (M. Kiesel), mougouie@dfki.de (B. Mougouie), baumann@dfki.de (S. Baumann), shankar.vembu@dfki.de (S. Vembu), romanelli@dfki.de (M. Romanelli), buitelaar@dfki.de (P. Buitelaar), engel@dfki.de (R. Engel), sonntag@dfki.de (D. Sonntag), reithinger@dfki.de (N. Reithinger), loos@eml.org (B. Loos), Hans-Peter.Zorn@eml-d.villa-bosch.de (H.-P. Zorn), micelli@eml.org (V. Micelli), porzel@tzi.de (R. Porzel), schmidt@ontoprise.de (C. Schmidt), weiten@ontoprise.de (M. Weiten), felix.burkhardt@t-systems.com (F. Burkhardt), jianshen.zhou@t-systems.com (J. Zhou).

1. Introduction

Providing good and substantial ontologies which stand the test of large application scenarios is a current bottleneck in Semantic Web research and application development. In theory, the Semantic Web should simply enable us to find ontologies on the WWW as needed, do some minor adjustments, integrate them, and use them within any desired application scenario. In practice, however, things are not that simple. The number of ontologies available is still rather small, and well-designed ontologies are as rare as substantial ones. Ontologies also differ with respect to their underlying ontological commitments and choices, which depend on the knowledge, expertise and skills of the engineers and also on the application scenarios for which they were originally developed. Consequently, even if usable and suitable domain ontologies are found on the WWW, their integration poses difficulties which are not readily solved from scratch, but rather call for a careful approach and the making of subtle design and methodological choices.

The SmartWeb project[1] rests heavily on the representation of knowledge by means of several substantial domain ontologies.[2] SmartWeb uses the broad expertise of the consortium in order to provide a single integrated demonstrator system which combines intelligent multimodal and mobile user interface technology with question-answering functionalities over both the open internet and specific thematic domains at the same time.

In order to realize this demonstrator, SmartWeb depends on several substantial domain ontologies for knowledge representation and reasoning. The main topics to be covered by these ontologies are the following:

- *Sport events*. The Football World Cup 2006 is the main application scenario for the SmartWeb system, and corresponding knowledge is modeled in the *S*port Event ontology.
- *Navigation*. SmartWeb user interfaces are mobile, e.g., by means of portable data assistants or by integration in automotive vehicles. Navigation when driving or walking is one of the core functionalities provided by the SmartWeb system.
- *Discourse*. Multimodal access to the WWW is one of the core features of the SmartWeb system. It is therefore necessary to model user interaction in a generic way, and this is done by means of the discourse ontology.
- *Multimedia data*. The SmartWeb system will be able to present multimedia data such as live streams or pictures when prompted to do so. This data is described by means of the Webcam and the SmartMedia ontologies.
- *Linguistic information*. This is needed for the support of ontology-based information extraction from text and semi-structured data.

Each of the domain ontologies just described may be used in several parts of the SmartWeb demonstrator system. They need to be interoperable and therefore need to be integrated into a single concise knowledge base. Domain knowledge, however, is distributed across the consortium, hence a centralized design and the engineering of a monolithic, all-encompassing ontology is inferior to a modular approach, where domain experts carry the main responsibility for their domain ontologies. This latter approach, however, raises the need for a common *modeling basis*, providing *conceptual clarity* of terms, and *ontology design patterns* for modeling consistency.

The methodological approach adopted by the SmartWeb consortium to address this need was to employ an abstract foundational ontology as a means to facilitate domain ontology integration. Domain experts are requested to align their domain ontologies to the foundational ontology, thus avoiding common pitfalls and in order to disambiguate meanings and to explicate implicit design decisions from the very start. The hypothesis was that the integration of the resulting domain ontologies should this way be relatively easy. After the integration, a major part of the foundational ontology which does not affect system behavior can then be removed before online usage, with the further advantage that an expressive ontology language can be used for the foundational ontology without having any negative effect on the eventual system performance.

Within SmartWeb, ontology experts were required to work hand-in-hand with domain experts with little theoretical or hands-on experience in ontology engineering. This paper is thus not only about a substantial and applicable ontology used in a major project, but at the same time also a report on a collaborative ontology engineering effort, a story about the usefulness of foundational ontologies, and about lessons learned in the process.

The paper is structured as follows. In Section 2 we discuss the issue of a suitable foundational ontology for SmartWeb, and why we opted for a hybrid solution engineered from the DOLCE and the SUMO ontologies. In Section 3 we discuss in more detail the resulting SmartSUMO ontology, and the process which led to it. In Section 4 we describe the aforementioned domain ontologies which are aligned by means of SmartSUMO, resulting in the combined SmartWeb Integrated Ontology (SWIntO). Section 5 gives a more detailed account of application aspects. We conclude in Section 6. The SmartWeb Integrated Ontology is available from http://www.smartweb-project.de/ontology.html.

2. Foundational ontology

This section argues for the usage of a foundational ontology as a modeling basis for the domain ontologies of SmartWeb (Section 2.1). We discuss the process of choosing the right foundational ontology in Section 2.2. By comparing the most promising foundational ontologies on the basis of ontological choices, we can narrow the choice down to DOLCE and SUMO.

[1] http://www.smartweb-project.org.

[2] SmartWeb is a large international project funded by the German Ministry for Education and Research (BMBF), running from 2004 to 2007. Its consortium, led by scientific director Wolfgang Wahlster, consists of 15 partners from universities, research centers, and research divisions of several companies. It combines expert knowledge from such diverse areas as intelligent user interfaces, mobile technologies and devices, automotive industry, Web service development and providers, machine learning, computational linguistics, knowledge extraction, multimedia data analysis, knowledge management, and semantic technologies.

Both foundational ontologies are further discussed in Sections 2.3 and 2.4. We conclude that a combination of both would fit the needs of the project best.

2.1. *Why use a foundational ontology?*

SmartWeb requires the representation of diverse information such as natural language processing data, Web services annotation, geographic information for navigation, etc. Ontologies are usually considered as an answer to such needs for information integration. If ontologies are to be used only for semantic access to a specific resource within a given community, the intended meaning of the terms used within the community is generally known in advance by all its members. The ontology can therefore be limited to only describing relevant structural relationships among relevant terms, i.e., primarily taxonomic relationships.

On the flip side, this does mean that the meaning of the terms may be loose or ambiguous. This can pose a problem when multiple independently developed ontologies must be integrated, as is the case in the SmartWeb system. For the particular situation of the SmartWeb project, where a large distributed team of domain experts is cooperating, terms must be defined very precisely (with explicit ontological commitment) to avoid terminological and conceptual ambiguities during integration. This requires a rich axiomatization of the ontologies as well as adequate informal documentation.

To address this issue, we use a foundational ontology for SmartWeb. A foundational ontology is an axiomatic theory about the high-level domain-independent categories in the real world, such as object, attribute, event, spatial and temporal connections, etc. Below we give the major advantages that promise a fruitful usage of a foundational ontology for SmartWeb.

Modeling basis: Foundational ontologies provide a starting point for building new ontologies. Instead of modeling from scratch, using a foundational ontology provides us with a predefined set of ontological entities that we can reuse for the domain ontologies of SmartWeb.
Conceptual clarity: Foundational ontologies provide a reference point for rigorous comparisons among different possible ontological approaches, and a framework for analyzing, harmonizing, and integrating existing ontologies and meta-data standards.
Ontology design patterns: In an ideal case, a foundational ontology defines ontology design patterns for re-occurring modeling needs, such as location in space and time, that we might apply for superior design and modeling consistency.

The advantages have to be traded-off against the additional efforts of familiarizing with the foundational ontology. Understanding foundational ontologies is not always easy because of their abstract nature. Often, philosophical background is required. However, for SmartWeb, we concluded that the advantages paid off. Problems with respect to information integration would have surfaced painfully in the later stages of the project otherwise.

2.2. *The right choice*

In the previous section we have made evident that the usage of a foundational ontology is crucial for a project such as SmartWeb. The natural next step is choosing the best fitting foundational ontology from about a dozen freely available ones worldwide (an overview is given in ref. [27]). We devoted an extra project report to this subject (cf. [8]) where we discuss the most promising candidates, viz., BFO, DOLCE, OCHRE, OpenCyc, and SUMO in light of ontological choices. A well-engineered foundational ontology is very specific about the ontological choices to which it commits. Hence, we are prompted to decide whether the ontological choices are suitable for our purposes. We discuss typical ontological choices, which are also called ontology meta-criteria, in the following paragraphs (cf. [24] for a detailed discussion).

Descriptive vs. revisionary: A *descriptive ontology* aims at describing the ontological assumptions behind language and cognition by taking the surface structure of natural language and common sense seriously. The distinction between things and events is typically considered as a human perception and is adopted by descriptive ontologies. A *revisionary ontology* is committed to capture the intrinsic nature of the world. As a consequence, an ontology of this type may impose that only entities extended in space and time exist.
Multiplicative vs. reductionist: A *multiplicative ontology* aims at giving a reliable account of reality as it allows different entities to be co-localized in the same spatio-temporal coordinate. These co-localized entities are assumed to be different because they have incompatible essential properties. A *reductionist ontology* postulates that each spatio-temporal location contains at most one object: incompatible essential properties are regarded as being linked to different points of view from which one can look at the same spatio-temporal entity.
Possibilism vs. actualism: The fundamental thesis of *actualism* is: “Everything that exists is actual.” *Possibilism* is the denial of this thesis and there are various forms of possibilism that correspond to the various ways in which one can deny this thesis. Many of our reflective and creative thoughts seem to be about possibilities and much of our logical reasoning involves drawing conclusions which, in some sense, necessarily follow from premises that we already believe. When committing to possibilism, we are able to represent possibilia, i.e., possible entities, in our domain. In this case, the representation language is required to express modalities, i.e., quantification over worlds. Typically this coincides with using a modal logic.
Endurantism vs. perdurantism: A fundamental ontological choice deals with the notion of change. What does it mean for an entity to change? This question raises the problem of variation in time and the related issue of the identity of the objects of experience. There are two main approaches, viz., *endurantism* (also called 3D paradigm) and *perdurantism* (also called 4D paradigm). Perdurantism assumes that entities extend in time and in space. That means entities have both spatial and temporal parts (and, therefore, four dimensions). Endurantism treats entities as 3D objects (sometimes called endurants or contin-

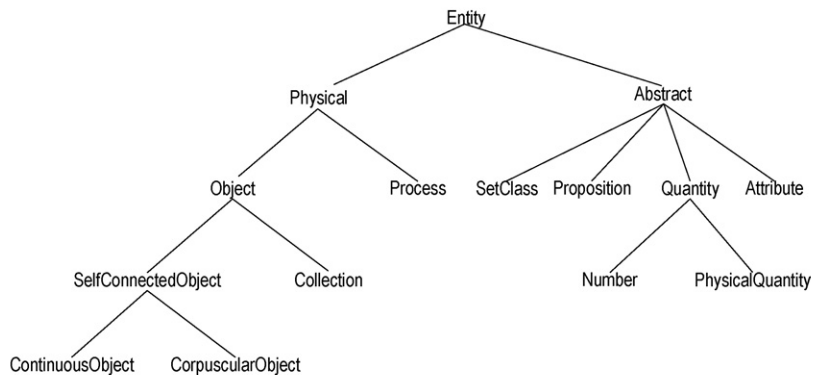

Fig. 1. SUMO taxonomy.

Table 1
Foundational ontologies and their ontological choices [8]

Requirement\ alternative	BFO	DOLCE	OCHRE	OpenCyc	SUMO
Descriptive	–	×	–	×	×
Multiplicative	–	×	Unclear	Unclear	×
Actualism	×	–	–	Unclear	Unclear
4D	×	×	–	Unclear	×

uants) that pass through time and are wholly present at each point in time. Hence, 3D entities do not have temporal parts.

The SmartWeb system is targeted at the end-user and will thus model artifacts of human common sense. Therefore, the SmartWeb foundational ontology should be *descriptive*. *Multiplicism* is a consequence because it is usually nearer at human common sense than reductionist modeling. Regarding the choice between possibilism and *actualism* we claim that the latter will be more practicable for the ontology engineers of the project because modalities would raise the complexity of modeling. Finally, the modeling of *4D* entities is required, e.g., for navigational concepts, such as `Translocation`, whose temporal parts would be `Maneuvers`.

In ref. [8], we have narrowed the choice of suitable ontologies for SmartWeb down to two: DOLCE and SUMO. In Table 1 we can see that both ontologies meet most requirements. The decision between DOLCE and SUMO boils down to one between conceptual clarity and easy accessibility, and to the assessment of possible long-term benefits which may be obtained from using DOLCE. In the following sections we discuss DOLCE and SUMO in more detail.

2.3. SUMO

The Suggested Upper Merged Ontology (SUMO)[3] is the most prominent proposal under consideration by the IEEE Standard Upper Ontology (SUO) working group[4] [26]. It is an attempt to link categories and relations coming from different top-level ontologies in order to improve interoperability, communication and search in the Semantic Web area. The development of SUMO was based on the merging of different ontology modules and theories: John Sowa's upper level ontology [36], Russell and Norvig's upper level ontology [32], James Allen's temporal axioms [1], Casati and Varzi's formal theory of holes [10], Barry Smith's ontology of boundaries [34], Nicola Guarino's formal mereotopology [4] and various formal representations of plans and processes, including the Core Plan Representation (CPR) [29] and the Process Specification Language (PSL) [17].

A sketch of the taxonomy is depicted in Fig. 1. The topmost concept in SUMO is `Entity`, which is further split into `Physical` and `Abstract`. `Physical` entities are further divided into `Objects` and `Processes`. Other general topics, which are not shown in Fig. 1, include: structural concepts (instance, sub-concept), general types of objects and processes, abstractions (including set theory, attributes, and relations, number, measures, temporal concepts, such as duration and parts and wholes) [31]. The taxonomy is large and even features many domain concepts, such as `Hotel` or `Organization`.

Because of its characteristic merging of different ontology modules and theories, SUMO is actually not influenced by any specific theoretical approach. Rather, it tends to adopt the general categories from various ontology proposals. In this context, we should say that SUMO does not clearly adopt either a *multiplicative* or a *reductionist* approach. That is, the major part of its theories commits to a *multiplicative* stance. We encounter the same dilemma regarding the choices *possibilism* vs. *actualism*, as well as *endurantism* vs. *perdurantism*. We classify SUMO as being *descriptive* because it adopts the common sense distinction between objects and processes.

SUMO provides quite a rich axiomatization formalized in the Standard Upper Ontology Knowledge Interchange Format (SUO-KIF), a variation and simplification of the Knowledge Interchange Format (KIF) [14], and in OWL Full.[5] However, the axiomatization suffers from several shortcomings. A lot of information is represented as instances whereas other modules use concepts on the same level, concepts are instances at the same time, relations are instantiated between concepts, and some relations are even modeled as concepts (e.g., there is a con-

[3] http://ontology.teknowledge.com/.
[4] http://suo.ieee.org.
[5] http://www.w3.org/2004/OWL/.

cept BinaryRelation). Furthermore, all SUMO versions come in one monolithic file with about 15,000 lines. The above-mentioned modeling flaws make it indeed very hard to work with SUMO. In addition, it is not suitable for the required reference purposes due to its lack of ontological commitment. However, SUMO provides a rich taxonomy that can be applied fruitfully for the domain ontologies of the project.

2.4. DOLCE

DOLCE belongs to the WonderWeb library of foundational ontologies [23]. It is intended to act as a starting point for comparing and elucidating the relationships with other ontologies of the library and also for clarifying the hidden assumptions underlying existing ontologies or linguistic resources such as WordNet [22]. It has been successfully applied in different domains, such as law [18], biomedicine [13] and agriculture [15].

Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) is based on the fundamental distinction between enduring and perduring entities. The main relation between Endurants (i.e., objects or substances) and Perdurants (i.e., events or processes) is that of participation: an Endurant "lives" in time by participating in a Perdurant. For example, a natural person, which is an Endurant, participates in his or her life, which is a Perdurant. DOLCE introduces Qualities as another category that can be seen as the basic entities we can perceive or measure: shapes, colors, sizes, sounds, smells, as well as weights, lengths or electrical charges. Spatial locations (i.e., a special kind of physical quality) and temporal qualities encode the spatio-temporal attributes of objects or events. Finally, Abstracts do not have spatial or temporal qualities and they are not qualities themselves. An example is Regions used to encode the measurement of qualities as conventionalized in some metric or conceptual space. The basic concept hierarchy is sketched in Fig. 2.

As reflected by its name, DOLCE has a clear *descriptive* bias, in the sense that it aims at capturing the ontological categories underlying natural language and human common sense. DOLCE embraces the *multiplicative* approach: starting from the observation that one tends to associate objects to incompatible essential properties, DOLCE provides a clear and detailed treatment of objects and properties assuming that different entities can be co-located in the same space-time. DOLCE allows modeling 3D objects, i.e., Endurants, as well as 4D objects, i.e., Perdurants. Thus, it commits to both *endurantism* and *perdurantism*.

DOLCE features a rich reference axiomatization in modal logic S5, thereby committing to *possibilism*. The axiomatization captures ontology design patterns such as location in space and time, dependence or part-hood. Its core is minimal in that it only includes the most general concepts and patterns. This makes it well-suited for modularization. In fact, there is a wealth of additional theories that provide additional ontology design patterns and can be included on demand. Examples are Descriptions & Situations for contextualization, the Ontology of Plans, the Ontology of Time, or the Ontology of Information Objects [12].

We can say that DOLCE is conceptually sound, and thus ideally suited for reference purposes. However, it does not provide an extensive and detailed taxonomy like SUMO. Besides, SmartWeb participants were actually afraid of using DOLCE because of its abstract nature. Hence, we have opted for an integration of both DOLCE and SUMO that fruitfully combines their advantages. We discuss this integration in the following section.

3. SmartSUMO

In the previous section, we discussed the need for a foundational ontology and identified the integration of DOLCE and

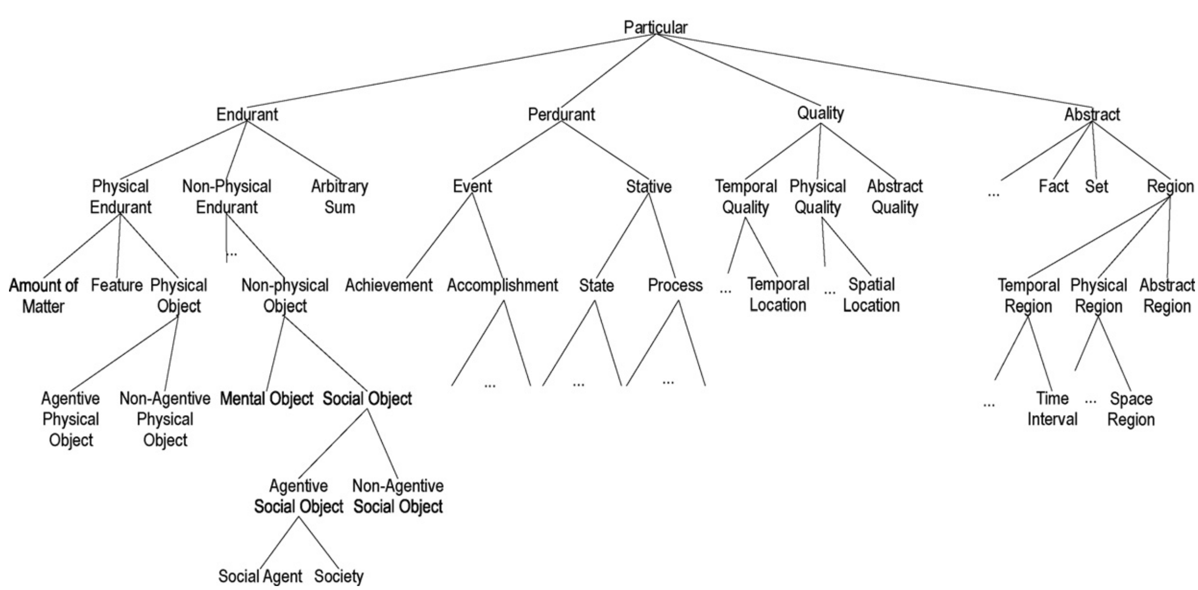

Fig. 2. DOLCE taxonomy.

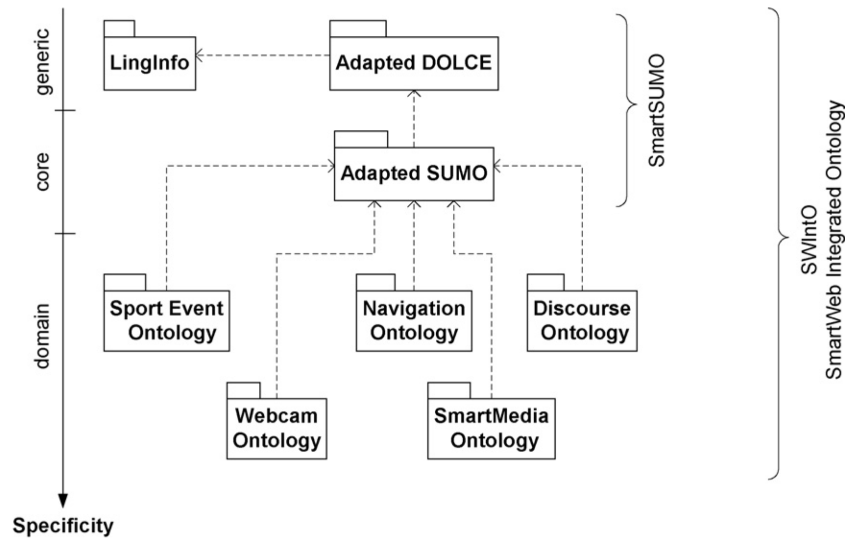

Fig. 3. Overview of SmartSUMO/SWIntO as UML package diagram. Packages represent ontology modules; dotted lines represent dependencies between modules. An ontology O_1 depends on O_2 if it specializes concepts of O_2, has relations with domains and ranges to O_2 or reuses its axioms.

SUMO as the best available choice. In this section, we show how this integration has been achieved. Both ontologies had to be adapted and extended in order to arrive at the integrated ontology, called *SmartSUMO*. These modifications are discussed in Sections 3.1 and 3.2 for both SUMO and DOLCE, respectively. Furthermore, we discuss an orthogonal extension of SmartSUMO for linguistic annotation, called LingInfo, in Section 3.3.

Fig. 3 shows an overview of SmartSUMO and the SmartWeb domain ontologies (the integrated ontology is called SWIntO—the SmartWeb Integrated Ontology). SmartSUMO's role is to act as a sound-modeling basis for building the SmartWeb domain ontologies, providing a rich taxonomy, ontology design patterns and predefined axioms. These domain ontologies are discussed in Section 4. We refer to the modified version of SUMO as a *core* ontology because we preserve its rich taxonomy to a large extent. The taxonomy features specific concepts such as `Hotel` or `Organization` (cf. [27] for further reading on the classification).

3.1. Modifications to SUMO

When constructing the SmartSUMO ontology, the majority of modifications and extensions were made to the SUMO ontology. As the first step, we obtained a version of SUMO that can be loaded and edited in common ontology editors, such as Protégé,[6] OilEd [5], or OntoEdit [33]. We took the OWL Full version as a starting point and basically reduced the axiomatization to a simple RDFS[7] level. We removed instances, concepts that were declared as instances of other concepts, relations that were instantiated between concepts, and some relations that were modeled as concepts. These steps were necessary to make SUMO conform to DOLCE because the latter limits modeling to the schema level. The resulting version contained about 600 concepts with about 250 relations with domain and range restrictions.

The second step comprised the actual alignment of the SUMO taxonomy to DOLCE. As discussed in Section 2.4, the DOLCE taxonomy features a well-designed minimal core of generic concepts (cf. Fig. 2) with about 350 concepts and 150 relations. Thus, to align SUMO to DOLCE, we pruned the upper-level of the SUMO taxonomy and aligned the remaining concepts to appropriate DOLCE categories. A sketch of this alignment is depicted in Fig. 4. During the alignment, it became apparent that grasping the intended meaning of SUMO's terms is quite difficult because of the loose merging of several theories in SUMO. Finding the best fitting super-concept in DOLCE for a SUMO term was therefore non-trivial. In addition, DOLCE design patterns such as the one for modeling qualities of endurants via regions, had to be taken into consideration when performing the alignment. Therefore, SUMO concepts, such as `Temperature` and `Length`, became sub-concepts of DOLCE's `Physical-Region`.

The third and final step consisted of several smaller modifications performed iteratively over time. For example, we encountered difficulties with the use of multiple inheritance in SUMO. The concept `Hotel` was declared both as sub-concept of `ResidentialBuilding` and `CommercialAgent`. While this holds naturally in the way we think about hotels, within our SmartWeb context, it is an unnecessary ambiguity, since we expect that the primary meaning of `Hotel` will be that of a building rather than that of a company. We therefore resolved the ambiguity by making `Hotel` a sub-concept of `ResidentialBuilding` only. The latter is a `StationaryArtifact` and, hence, a DOLCE `NonPhysicalObject`. We encountered and resolved similar ambiguities with terms such as `Country`, `City`, or `State`, whose meanings can be both non-physical, geopolitical objects, as well as physical places.

[6] http://protege.stanford.edu.
[7] http://www.w3.org/RDF/.

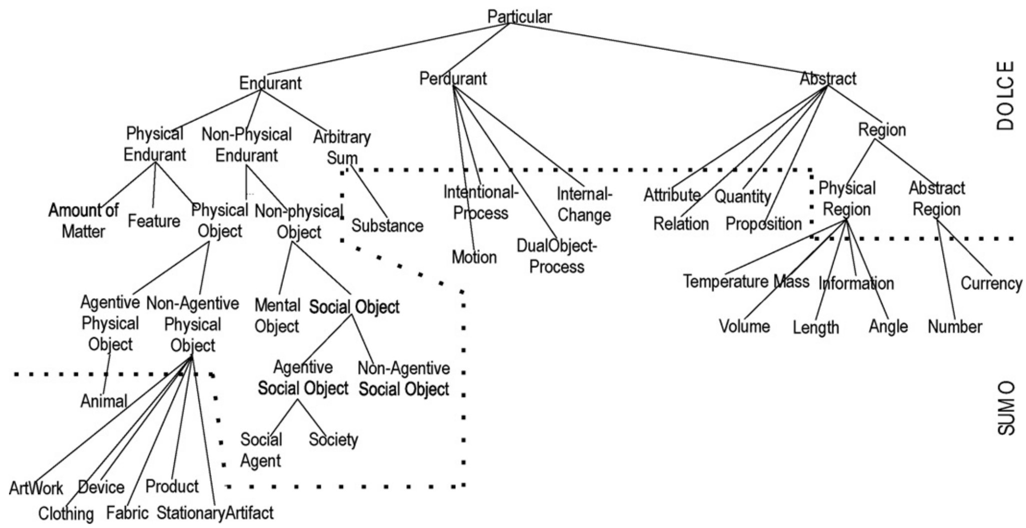

Fig. 4. Sketch of the DOLCE-SUMO alignment.

A major extension worth discussing in some detail is the agreement on how to model addresses. After initial modeling efforts, it became apparent that this information is required in all domain ontologies and therefore needed a unified model. We opted for a flexible solution where Smart-SUMO 's `Address` points to its related `Country`, `City`, `Road`, and `Geoposition`. Each of the concepts of course features further relations and information. Similarly, we also added instances of all countries and German cities, because they are required by the Navigation and Sport Event ontologies.

The SmartSUMO ontology evolved after several meetings with the domain ontology experts. Countless smaller modifications and extensions were introduced over time. Typically, common modeling needs of two or more domain ontologies were placed in SmartSUMO. For instance, at one stage, we added a means to model the project's Web services as discussed in Section 5.3. The SUMO part of SmartSUMO has eventually decreased to a minimum because of the above-mentioned problems. The concise and abstract nature of DOLCE proved to be rather useful and initial fears of SmartWeb participants were alleviated.

3.2. *Modifications to DOLCE*

The DOLCE portion of SmartSUMO, called *SmartDOLCE*, required relatively minor modifications and extensions, partly because of its foundational nature. The primary difficulty in using DOLCE proved to be the concepts beneath `Perdurant`, i.e., `Stative` and `Event`. It was too difficult for ontology engineers to understand the intended meaning of these terms and to classify their own perdurants underneath them. Hence, we only kept the concept `Perdurant` and made SUMO's `Processes` direct sub-concepts.

Among the extensions to DOLCE, the modeling of time is particularly worth mentioning. DOLCE is quite unspecific with respect to a concrete modeling of time-points in terms of hours, minutes, and seconds. There are only the basic patterns between `Endurants`, `Perdurants`, and `TemporalRegions` (these are the axioms called `presentAt` and `happensAt`). Therefore, we refined the patterns and introduced concrete `TemporalRegions` with all the necessary attributes: `TimePoint`, `TimePointRelative`, and `TimeInterval`.

Another extension worth mentioning is the common treatment of names. During the project, it became evident that every entity can have different kinds of names and abbreviations. Hence, we introduced our own *Denomination* ontology design pattern as depicted in Fig. 5. We applied this pattern in different domain ontologies, e.g., for modeling names of natural persons (cf. Fig. 5) and countries.

Finally, two modules of DOLCE were added to SmartDOLCE: Descriptions & Situations and the Ontology of Information Objects (cf. [16,12] for more information on the ontologies). Descriptions & Situations (DnS) is an ontological theory of contexts that comes in the form of an ontology module. DnS can be considered an ontology design pattern for structuring core and domain ontologies that require contextualization. In Sections 4.2.2 and 5.2, we describe the important role of well-

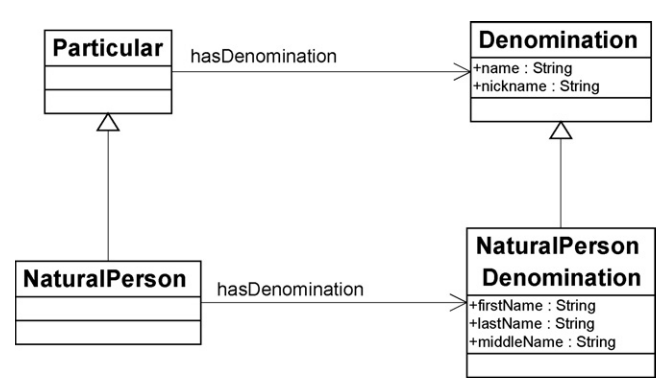

Fig. 5. The SmartWeb *Denomination* ontology design pattern applied for natural persons and sketched as UML class diagram. Classes represent concepts, associations represent relations. Inheritance is interpreted as the *isa* relation.

founded and explicit models of contextual knowledge in mobile and open domain systems, such as SmartWeb.

The Ontology of Information Objects provides an ontology design pattern that allows us to concisely model the relationship between entities in an information system and physical entities. This is required for the Webcam Ontology later on (cf. Section 4.4).

3.3. *Representation of linguistic information with LingInfo*

The DOLCE and SUMO parts of SmartSUMO are mostly directed at the *referent* side of the well-known 'meaning triangle' [28], but much less at the *symbol* side. To allow for automatic multilingual knowledge markup a richer representation of the linguistic symbols is required for ontology entities. Such information is mostly missing or represented only in a very impoverished way, leaving the semantic information in SmartSUMO without a grounding to the linguistic domain.

As the SmartWeb system heavily relies on linguistic information, e.g., for natural language processing, we introduced a multilingual lexicon model, called LingInfo. LingInfo allows the representation of linguistic symbols for ontology entities (see also refs. [7,3]). Linguistic symbols, i.e., simple words or more complex terms, are represented in LingInfo in order to provide the meaning of these words or terms with a more or less extensive representation of their linguistic features, e.g., if the word is a noun or a verb, if it is atomic or can be split into multiple words, etc. Linguistic information consists of term variants – for each language covered by the ontology – with lexical and context information for each term:

- *Language-ID*: ISO-based unique identifier for the language of each term.
- *Part-of-speech*: (possibly ISO-based) representation of the part of speech of the head of the term.
- *Morphological decomposition*: representation of the morphological structure (segments, head, modifiers) of a term.
- *Syntactic decomposition*: representation of the syntactic structure (segments, head, modifiers) of a term.
- *Statistical and/or grammatical context model*: representation of the linguistic context of a term in the form of N-grams, grammar rules or otherwise.

To represent terminology in different languages we created an extension of RDF-based domain knowledge representation with the meta-class `Class-WithFeats`.

Fig. 6 depicts the part of LingInfo in detail that deals with the representation of linguistic features, which is mainly the mor-

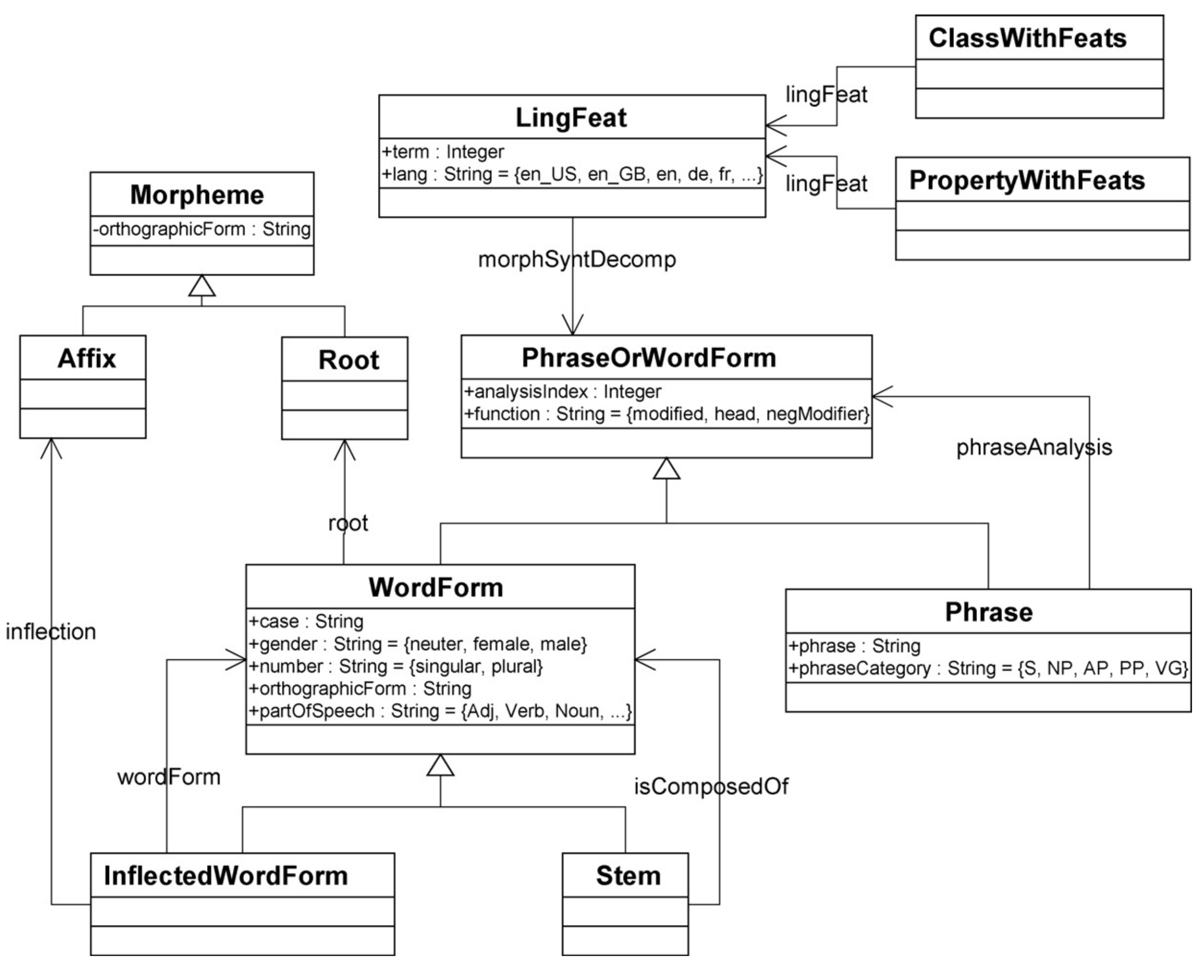

Fig. 6. Part of the LingInfo ontology depicted as UML class diagram.

phosyntactic decomposition of phrases and word forms down to stems, roots, morphemes, affixes, etc. Apart from having linguistic properties such as gender, number, part of speech, case, etc., word forms have the property semantics which is a back link into the ontology allowing semantics to be assigned to them.

4. Domain ontologies

As mentioned in Section 1, SmartWeb depends on several domain ontologies for knowledge representation and reasoning. Each of the ontologies is used in several parts of the SmartWeb demonstrator system. They need to be interoperable and therefore need to be integrated into a single, preferably concise, knowledge base. Domain knowledge, however, is distributed across the ontology building team. Therefore, a modular approach to ontology construction is required, where domain experts carry the main responsibility for their domain ontologies. In this section, we present the individual domain ontologies and show how they benefit from SmartSUMO providing a common *modeling basis*, *conceptual clarity* and *ontology design patterns*. Several group meetings were required over the course of the project to achieve a consistent and stable version of SWIntO.

4.1. Sport Event ontology

Since the questions to be answered by the SmartWeb system are primarily centered around football, the project relies on a Sport Event domain ontology that models football events of varying granularity (tournaments, matches and match events such as goal shots) as well as persons, places, and some more abstract entities like result tables. Although we cannot claim that it completely models the whole world of football, it provides a structure for almost all football-related information in databases, web pages and books. The Sport Event ontology is based on the SmartSUMO foundational ontology, aligning its concepts to the SmartSUMO ontology. Thus, the Sport Event ontology benefits from SmartSUMO's *conceptual clarity*. Furthermore, the Sport Event ontology applies some of SmartSUMO's *ontology design patterns* for consistent modeling. The Sport Event ontology features a large set of instances that primarily model facts of the Football World Cups 1930–2006, serving not only as a basis for SmartWeb's ontology-based search components but also as validation for the modeling decisions.

The main modeling challenges when creating the Sport Event ontology were proper multilingual handling of names of instances, giving structure to the loosely defined terms in the sports domain, and providing linguistic information for concepts and relations, thus allowing extraction components to automatically create new fact instances and even facilitating semi-automatic ontology extensions. Not surprisingly, there were also several technical challenges: keeping the Sport Event ontology synchronized with SmartSUMO was difficult at times, modifying components that build on the Sport Event ontology, such as the query generation modules, and finally making sure that the facts supplied by automatic components fit to the ontology schema proved difficult. However, a detailed discussion of these issues is out of scope for this paper.

In order to give an idea about the size of the Sport Event ontology, there are more than 500 concepts, about 70 relations, about 2000 linguistic annotations in the Sport Event ontology, not including relations and annotations inherited from SmartSUMO.

4.1.1. Overview of the Sport Event ontology

To present the essential features of the Sport Event ontology, we start with the concept `FIFAWorldCup` (sub-concept of `FootballWorldCup`), shown in Fig. 7, along with two of its instances and example instances of the concepts `WorldCupAnthem`, `WorldCupMascot` and `OfficialWorldCupBall`.

In Fig. 8, an overview of the concept structure surrounding `FIFAWorldCup` is given. Since RDF(S) only supports one level of instance-of relationships, we can only use concepts and instances (meta-classes notwithstanding), forcing users to draw a sharp and sometimes artificial border between the two, as can be seen in the figure. One could argue that, for example, `UEFAChampionsLeague` is an instance of `EuropeanFootballTournament`, rather than a sub-concept as we have modeled it. However, we chose to do so to leave the instance layer for instances of `UEFAChampionsLeague` such as a league taking place in a certain year. We also make use of multiple inheritance, as can be seen in the figure, because it is naturally required as a consequence of our concept vs.

Fig. 7. Example for an `FIFAWorldCup` instance as UML object diagram. Concept instances are represented by objects and instantiated relations by object associations.

Fig. 8. `smartsumo:Contest` and some of its subclasses.

instance design decision. If we had decided to draw the concept/instance border higher up the hierarchy, modeling concepts such as `UEFAChampionsLeague` as instances, this would have two major drawbacks: first, the instance would have to be an instance of two concepts (`EuropeanFT` and `BetweenClubsFT`), which is supported by only a few tools, and second, the relationship between `UEFAChampionsLeague` and its annual instances would have to use an arbitrary relation, not taking advantage of the RDFS built-in semantics.

There are a number of other important concepts, most notably the `Match` and `Team` concepts and their sub-concepts, for modeling individual (football) matches, football teams, and clubs. An instance of a football `Match` can contain information on the stadium and country it was held in, its participants (both players and other people such as referees and trainers), events that occurred within the match, and other information such as the number of spectators. Modeling teams, on the other hand, was quite a challenge since `Team` is a quite ambiguous concept, ranging from football clubs and squads to the team playing in an actual match.

4.1.2. Sport Event ontology design patterns

There are certain relations in the Sport Event ontology which play important roles. We take a look at some of them in the following.

- `dolce:hasDenomination`: The SmartSUMO ontology design pattern called `hasDenomination` is applied and refined to capture first, middle, last, or nickname of a person.[8] As an example, the relation `dolce:hasDenomination` of *Roberto Carlos* of `smartsumo:Man` refers to the instance *Roberto Carlos* of `dolce:NaturalPersonDenomination` which contains the information such as first, last, and nickname of Roberto Carlos. This way, it is possible to model his complete name, *Roberto Carlos da Silva*.
- `impersonatedBy`: Let us consider the football player Roberto Carlos. Although he is a famous football player, few people know his real name, since Brazilian football players often have nicknames. Their real name is less important to most football fans, even though it is quite important to him as a person. Assuming Roberto Carlos is also his real name, the instance *Roberto Carlos* of `FootballPlayer` refers to the football player with his sports-related information and the instance of a person who impersonates this player is *Roberto Carlos* of `smart-sumo:Man`. The relation `impersonatedBy` is therefore used to relate these two instances.
- `hasUpperRole`: Within a team, a football player may play different roles from match to match. For instance, he might be the captain in one match but not in another one. He might get substituted, injured, booked, sent off in a match, or even have different shirt numbers in different matches (Roberto Carlos wears shirt number 3 in his club Real Madrid and 6 in Brazilian national team). *Roberto Carlos* (a `FieldMatchFootballPlayer` instance used for modeling the data associated to Roberto Carlos participating in a game) is linked to *Roberto Carlos* (instance of `FootballPlayer`) using the relation `hasUpperRole`.

4.2. Navigation Ontology

The domain of human navigation represents a central application area for mobile systems such as SmartWeb. In the SmartWeb Navigation Ontology, concepts and instances relevant to the domain are modeled and important links for possible contexts are provided. The goal here was not to build a topology for a country, in which the distance between two places, for instance, is described. This task is already solved by technologies such as the spatial database engine (SDE) and geographic information systems such as GIS ([37]). Our goal is to assist human users in wayfinding tasks.[9] The Navigation Ontology relies on SmartSUMO as a *modeling basis* and also applies one of SmartSUMO 's *ontology design patterns* as discussed in the subsections below.

4.2.1. Basic navigational concepts

The goal of the ontology is not only to model entities relevant for the navigation domain but also to model corresponding processes. Therefore, perdurant concepts for motion are modeled through `smartsumo:Translocation`. The alignment of the basic concepts to the SmartSUMO ontology is based on the fact that a navigator moves from one point to the other on a road with a means of transportation, influenced by the time, day and weather. Here, the Navigation Ontology benefits from SmartSUMO as a *modeling basis* as most of the corresponding concepts are provided by SmartSUMO:

1. Perdurant concepts for motion modeled through `smartsumo:Translocation`.
2. Endurant concepts for a navigator, such as roads, buildings and places as well as countries, cities and provinces.
3. Quality regions [23] for the user, for roads, weather and time.
4. Instances of cities, bars and hotels.

[8] It may be used for other types of entities such as countries, too.

[9] To give an idea of the navigation ontology's size: until now it consists of 519 direct concepts, 231 relations and 3289 instances.

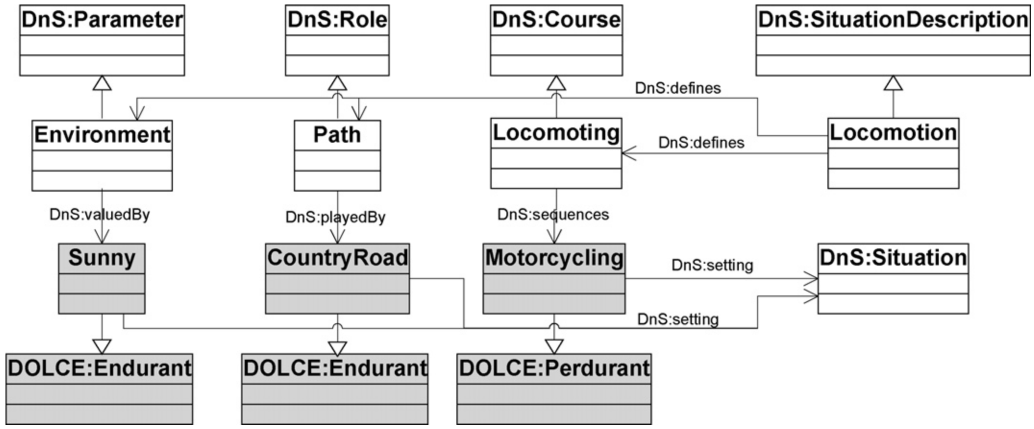

Fig. 9. The Descriptions & Situations ontology design pattern applied in the Navigation Ontology and depicted as UML class diagram. Grey classes represent the *ground* entities of DOLCE. *Descriptive entities* are `Parameters`, `Roles` and `Courses`.

The structure of the basic ontology is important, because the modeling of contexts with Descriptions & Situations is based on these concepts, as discussed in the next section.

4.2.2. *Basic navigational design patterns*

In mobile dialog systems, contextual information of the user is of high significance as the user expects up-to-date services. Furthermore, he navigates through a dynamically changing world (e.g., weather, time, and traffic), which makes the adequate representation of context knowledge essential in order to present relevant services to the user.

For example, the following situations demonstrate the necessity of including extra-linguistic situational knowledge for the domain of human navigation in real space [30]:

- A pedestrian might prefer public transportation over walking when it is raining, even for short distances.
- A motorcyclist might prefer to use winding country roads over interstate highways when it is warm and sunny, but not when road conditions are bad.
- A car driver might like to know about conditions on a given road before committing to it.

These situations demonstrate how taxonomic knowledge alone is not sufficient for assisting human users in navigational tasks. The solution here is to include pragmatic knowledge in the ontology [21].

For representing such information, SmartSUMO provides the *Descriptions & Situations* ontology design pattern. DnS allows to represent contexts and states of affairs, as described in Section 3.2. When Descriptions & Situations is used with DOLCE, the DOLCE entities are called *ground* entities and the newly introduced entities of Descriptions & Situations are called *descriptive* entities. We also visualize this distinction in Fig. 9. `Parameters`, `Roles` and `Courses` are the descriptive entities which are special kinds of `ConceptDescriptions` (a `dolce:NonAgentiveSocialObject`). The descriptive entities "describe" the ground entities in the following way: `Parameters` are `valuedBydolce:Regions`, `Roles` are `playedBydolce:Endurants` and `Courses` are `sequencedolce:Perdurants`. The descriptive entities are aggregated by a `SituationDescription`, representing the context, through the `defines` association.

Thus, a descriptive entity of a perdurant concept could, for example, be `Locomoting`, which could be realized as `Good-WeatherMotorcycling` (a specialization of the process `Motorcycling`), performed on a `CountryRoad` when the `Environment` is `Sunny`. Fig. 9 gives an idea of how this context-dependent information is realized with the DnS design pattern.

4.3. *Discourse ontology*

The discourse ontology models multimodal discourse information within a conversational interaction between a user and a Semantic Web knowledge base and provides a system-wide ontological representation for the acquired knowledge. Currently, the ontology comprises four different branches: (1) multimodal dialog management concepts, (2) question-answering-related ontological concepts, (3) dialog modeling concepts and (4) Human–Computer Interaction (HCI) concepts. Representing such information requires the *conceptual clarity* of SmartSUMO as we will learn in the following subsections.

4.3.1. *Multimodal dialog management*

A key goal of the discourse ontology is to ensure data exchange between components of a multimodal dialog system. For this purpose, we derived an RDFS ontology from the EMMA (Extensible MultiModal Annotation Markup Language)[10] W3C standard. The specification contains necessary administrative meta-information for multimodal interaction data, such as the type of interaction and time-stamps. Currently, the EMMA standard only covers the interpretation of speech recognition analysis, providing, for this purpose, the `emma:Interpretation` element which may additionally contain application-specific markup or an `emma:Lattice` element.

[10] http://www.w3.org/TR/emma/.

In order to also model the output-related components (e.g., a presentation manager), we introduced an extension to the EMMA standard, which explicitly represents results of the speech recognition analysis. This extension is specified by the namespace SWEMMA (SmartWeb EMMA). In the following we present the two most important SWEMMA extensions:

- `swemma:Result`: This tag is used to represent the result of the natural language understanding and generation process, and may occur as direct child of the EMMA tag. In the result tag, we permit the use of Speech Synthesis Markup Language (SSML) tags to control the synthesis parameters.
- `swemma:Status`: This tag delivers status information about the progress of query processing and the actual result content at different levels of processing. This information is coded by the status tag which in turn, together with the internal turn-ids and timing information, contains instances of the discourse ontology relevant to status information.

4.3.2. *Question-answering-related ontological concepts*

The system-user dialog is modeled by `swemma: Interpretation` and `swemma:Result`. Within the discourse ontology, the `swemma:Interpretation` is inherited by `discourse:Query` and `discourse:Result`.

- `discourse:Query`: The `discourse:Query` concept models the user query to the system. Queries have the form of partially filled SmartSUMO instances (see Fig. 10) to be completed by result information. Queries also include a `discourse:Focus` concept instance that models the focus of the user utterance, i.e., the ontological concept of the information the user is asking for. The `discourse:Query` is mapped to DOLCE as an `abstract-proposition` entity.
- `discourse:Result`: The `discourse:Result` references the information the user is asking for, and contains the answer. The answer is realized as a media object description as specified in the SmartMedia ontology based on the MPEG7 standard (see Section 4.5).

Both `discourse:Query` and `discourse:Result` exhibit a link to the question-answering concept `discourse:AnswerType` which is specialized to different types of answers (e.g., `LocationAnswer`, `PersonAnswer`, etc.) to ensure full coverage of all forms of simple answers [19]. If it appears in the context of a `Query`, the `AnswerType` expresses the type of answer expected in the result. In the context of a `Result`, the `AnswerType` represents a realized answer ready to be routed to the presentation tool. Fig. 11 depicts the hierarchy of the QA-related concepts.

4.3.3. *Dialog modeling*

To be able to handle discourse/dialog specific phenomena like resolution of elliptical or anaphoric references, resolution of relative to absolute time expressions, the discourse ontology defines a semantic representation.

- `DialogActs` model the intention of the dialog messages (illocutionary acts) during the communication process between system and user. Those messages are the user or system queries, the results, clarifications, greetings, acknowledgements, or the like. For the QA domain as a task-oriented dialog domain, these acts are different from other domains like

```
<rdf:RDF>
  <emma:Emma>
    ...
    <discourse:Query rdf:about="http://smartweb.org/ind#i4">
      <discourse:text rdf:datatype="http://www.w3.org/2001/2001/XMLSchema#string">
       wer war 1990 Weltmeister</discourse:text>
      <discourse:dialogueAct>
        <discourse:Question/>
      </discourse:dialogueAct>
      <discourse:focus>
        <sportevent:DivisionNationalTeam rdf:about="http://smartweb.org/ind#i5"/>
      </discourse:focus>
      <discourse:content>
        <sportevent:WorldCup>
          <sportevent:heldOn rdf:datatype=
            "http://www.w3.org/2001/XMLSchema#string"
            >1990</sportevent:heldOn>
          <sportevent:winner rdf:resource="http://smartweb.org/ind#i5"/>
        </sportevent:WorldCup>
      </discourse:content>
      <emma:confidence rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
       0.75</emma:confidence>
    </discourse:Query>
    ...
  </emma:Emma>
</rdf:RDF>
```

Fig. 10. Example for the use of the `Query` concept in the interpretation of the utterance "*wer war 1990 Weltmeister?*" (*Who was world champion in 1990?)*. The tag sportevent: content contains a partially filled ontology instance to be completed by result information.

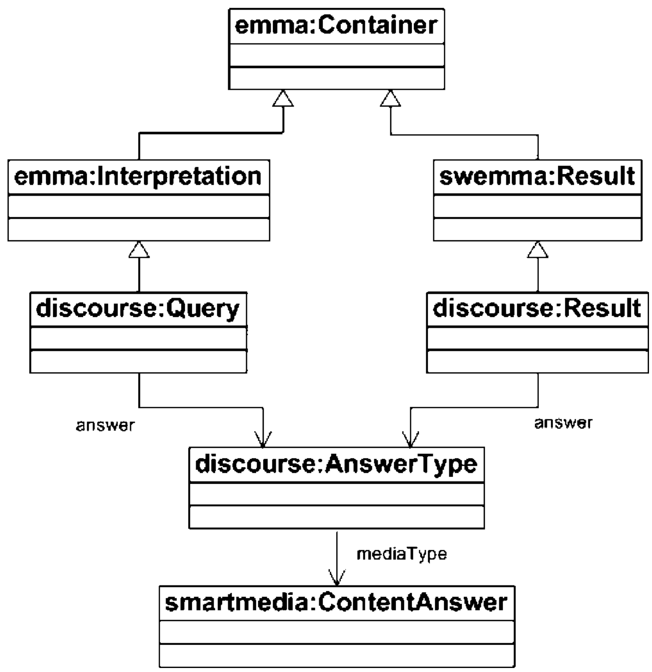

Fig. 11. The `discourse:Query` and `discourse:Result` concepts in relation to `discourse:AnswerTypes`. Classes represent concepts, associations represent relations. Inheritance is interpreted as the *isa* relation.

business appointment scheduling, or other problem solving activities, e.g., route directions.

- `RefProp` is used to resolve referential expressions in the context of discourse. The information whether a unit, e.g., a phrase, is definite or indefinite is mapped from the LingInfo (cf. Section 3.3) information onto the `RefProp` relation. This information is used in the inner dialog system communication. A unit labelled as definite indicates the presence of an anaphorical reference which has to be resolved. The information is passed to the discourse interpretation module which looks for the referenced item in the former user utterances.
- `TimeRepresentation`: For some temporal expressions, such as *on Sunday* or *in 3 days*, adequate concepts do not exist in the SmartSUMO ontology. The discourse ontology adds appropriate concepts. This allows having an explicit module within the dialog system to transform these temporal expressions to absolute time points and using instances of SWIntO as input and output representations.

4.3.4. HCI concepts

Human–Computer Interaction is usually understood as the discipline studying the design and requirements of computing systems interacting with humans. The question we address in the discourse ontology is how to represent different kinds of results (information design) and interaction metaphors (interaction and presentation design) which are created and presented to the user during the dialog [35]. The HCI concepts are coded into a pattern language for interaction design. The ontological representation shows its strength to achieve a common data model for all the HCI concepts and other discourse concepts. For example, we formulate patterns for incremental display of results and the layout of multimodal answers. With the help of the ontological descriptions we are able to express constraints on the combination of patterns: if a multimodal answer is to be presented, e.g., image and text, the incremental display must present the media instances both at the same time. It is still to be investigated which ontological pattern constraints are most useful for selection of particular pattern instances, and how they are best combined for a complete Human–Computer Interaction.

4.4. Webcam Ontology

The Webcam Ontology is used to model the connection between natural language sentences such as "*show me pictures of that area*" and the invocation of a Webcam-finder Web service that delivers a URL pointing to a videostream or an image-file given some geocoordinates. We started the modeling process by thinking up some competency questions, which can be divided into two categories:

- Natural language queries: Using a grammar as generator we created trigger phrases like "*show me pictures of the Brandenburg Gate*" or "*I'd like to see the soccer stadium*."
- Technical terms: Derived from Webcam definitions we defined technical terms related to Webcams, e.g., "picture resolution," "update interval" or "URL."

From the competency questions we derived concepts like `Image`, `Webcam`, `Camera`, or `Showing` and relations between them. As a next step we integrated the concepts into SmartSUMO. The *ontology design pattern* of information objects as provided by SmartSUMO proved to be very useful to model such concepts. Hence, we specialized and applied this pattern for a swift integration into SmartSUMO. Furthermore, SmartSUMO was required to act as a common *modeling basis* because many concepts were already used by other domain ontologies. For instance, the class for "Brandenburg Gate," `StationaryArtifact`, was used by the Navigation Ontology as well. Hence, we had to identify and transfer such concepts into SmartSUMO.

The core part of the resulting ontology is depicted in Fig. 12. The main connection between `Image` and `Camera` is given by the concept `Showing` which is a subclass of `Perdurant`. A second connection is given by the `produces` relation to `Camera`. `Image` as well as `Webcam` are modeled as `dolce:-InformationObjects`. `Webcams` and `DigitalCameras` can either be still-pictures or video in order to model streaming media. The connection between `Image` and instances of buildings such as *BrandenburgGate* is given by the `about` relationship inherited from `InformationObject`.

4.5. SmartMedia ontology

For the semantic annotation of multimedia content such as image, audio, video and text snippets we had to integrate MPEG-7. MPEG-7 is an ISO/IEC standard developed by Moving Picture Experts Group (MPEG) and it offers a comprehensive set of

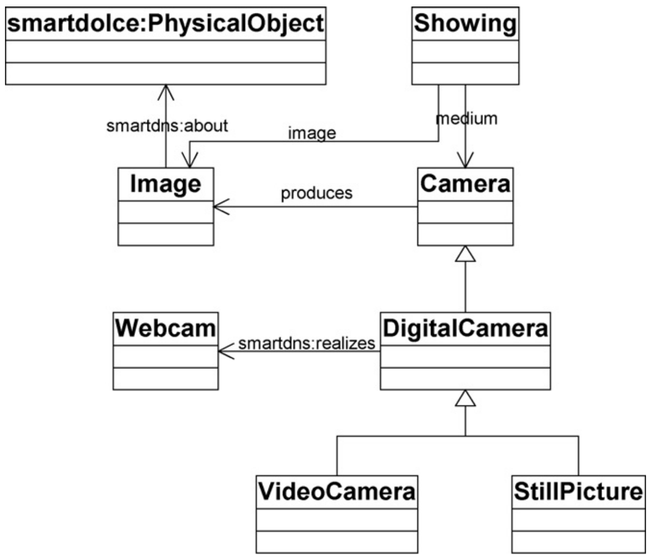

Fig. 12. Core of the Webcam Ontology model sketched as UML class diagram. Classes represent concepts, associations represent relations. Inheritance is interpreted as the *isa* relation.

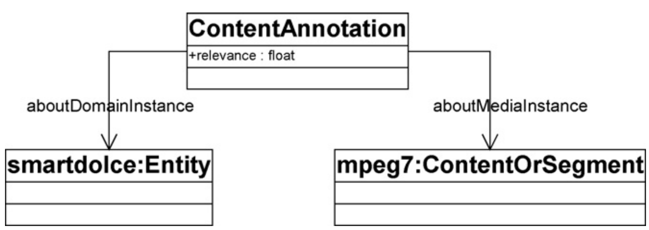

Fig. 13. `ContentAnnotation` is the uppermost concept of the SmartMedia ontology.

audio–visual description tools, which form the basis for applications to efficiently search and browse multimedia content.[11]

Even though the MPEG-7 framework provides tools for describing the high-level semantics of multimedia [6], they lack the expressiveness required to model the semantics of specific domains—for example, sports events. This shortcoming is commonly referred to as the *semantic gap*. We therefore decided to enrich the set of low-level descriptors provided by MPEG-7 with concepts from domain specific ontologies like Sport Event and link them to SmartSUMO for increasing *conceptual clarity*.

We designed a concept called `ContentAnnotation` (cf. Fig. 13) with relations to `Entity` from the DOLCE ontology and `ContentOrSegment` from MPEG-7. The attribute `relevance` indicates the strength of relationship between the domain instance and the MPEG-7 instance. For example, it takes a value of 1.0 for an image showing a specific football player (completely and exclusively) and a domain instance representing him. By linking concepts from MPEG-7 and domain specific ontologies, this concept forms the basis to bridge the *semantic gap*.

This integrated ontology is called SmartMedia ontology and can be used to describe not only the low-level multimedia concepts but also the high-level semantics as specified by the domain ontologies.

The MPEG-7 Descriptors[12] and Description Schemes[13] are specified in the XML Schema language. Hence, there is a lack of interoperability with the Semantic Web community, which uses languages like RDF Schema and OWL to represent ontologies. The first attempt to build an MPEG-7 ontology represented in RDF Schema was made by Hunter [20]. For the SmartWeb project, we decided to model relevant MPEG-7 concepts using ontology editors such as Protégé that also supports exporting ontologies into RDF Schema and OWL. In a first step, we modeled only the concepts that fit well to the project at hand instead of trying to use the entire specification.

The MPEG-7 Descriptor Schemes (DS) are organized into the following six categories: Basic Elements, Content Description, Content Management, Content Organization, Navigation and Access, and User Interaction (see ref. [20] for a brief description of each). We focussed on the Content Description and Content Management DS that suffice to model concepts describing storage features (such as format and encoding), spatial, temporal and spatio-temporal components (such as scene cuts, region segmentation and motion tracking), and low-level features (such as color, shape, texture, timbre and melody) of multimedia content.

The MPEG-7 concepts, namely `MultimediaContent` and `Segment`, make such kind of recursive modeling possible. We designed a concept called `ContentOrSegment`, which is an abstraction of the high-level entities `MultimediaContent` and `Segment` of MPEG-7 as shown in Fig. 14. This concept is the top-level concept of our MPEG-7 ontology. The top-level multimedia content entity is called `MultimediaContent`. The concepts of `Audio`, `Video`, `Image` and `Text` are sub-concepts of this top-level entity. As laid out in the MPEG-7 specification, these concepts are modeled in such a way as to describe the intrinsic recursiveness of multimedia in general. For example, an audio file can be considered to be a set of segments or snippets at different temporal locations, which could in turn be recursively divided. In the case of videos, the recursion also extends in the dimension of space giving rise to a complex set of temporal, spatial and spatio-temporal segments.

5. Applications of SWIntO

In this section, we present three *applications* within the SmartWeb system which make use of SWIntO. All three benefit from having a common *modeling basis*, *conceptual clarity*, and usage of *ontology design patterns* in SWIntO as a whole. Firstly, we discuss our solution for ontology-based information extraction. As an application for context-dependent utterance interpretation we then describe our application of Navigation Ontology in particular. Finally, we present our semantic integration of external Web services applying the ontology design pattern of information objects.

[11] http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

[12] Descriptors define the syntax and semantics of each feature.

[13] Description Schemes specify the structure and semantics of the relationships between components, which may be both Descriptors and Description Schemes.

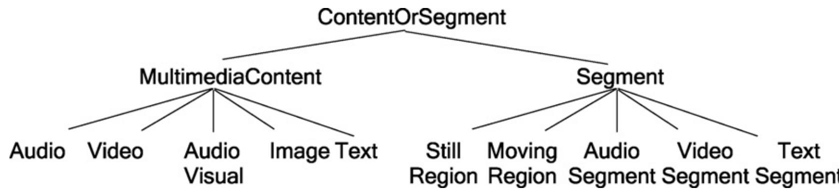

Fig. 14. MPEG-7 taxonomy.

5.1. *Ontology-based information extraction*

In this section, we describe the SmartWeb Ontology-Based Annotation (SOBA) system which makes use of the LingInfo ontology as an interface and additional knowledge repository [2]. As discussed in Section 3.3, the LingInfo component of the SmartWeb Integrated Ontology provides a rich information source for the linguistic annotation component, with which named entities and other, more complex structures are mapped to soccer-specific semantic structures as defined by the SWIntO ontology.

Without the *conceptual clarity* and strict use of *ontology design pattern*, the realization of the SOBA application would have been much more difficult. One of the crucial characteristics of SmartSUMO is that it provides a high level of abstraction allowing to model certain phenomena in a concise way. This is important here as the semantics of spatio-temporal prepositions such as *before*, *after*, or *above* can be specified in a well-defined and concise manner by resorting to SmartSUMO. Restrictions about which type of arguments a verb can take or an adjective can modify can in some cases also be appropriately specified with respect to SmartSUMO. To give an example, adjectives denoting a *color* such as 'green', 'red', 'blue', etc. can typically only modify a `dolce:PhysicalObject`.

The SmartWeb Ontology-Based Annotation system consists of a web crawler, linguistic annotation components and a component for the transformation of linguistic annotations into an ontology-based representation. The web crawler acts as a monitor on relevant web domains (e.g., the FIFA and UEFA web sites), automatically downloads relevant documents from them, and sends them to a linguistic annotation Web service. Linguistic annotation in the system is based on the *Heart-of-Gold* (HOG) architecture [9], which provides a uniform and flexible infrastructure for building multilingual applications that use semantics- and XML-based natural language processing components.

For the annotation of soccer game reports, for example, we use the Sprout named-entity recognition component in HOG with gazetteers, part-of-speech and morphological information [11]. Sprout combines finite-state techniques and unification-based algorithms. Fig. 15 shows a sample application of this part of the ontology, the decomposition of the German term "Fußballspielers" (meaning "of the football player"): *inst1* indicates that the term is an inflected word form (where the inflection is for forming the genitive) with stem "Fußballspieler" (meaning "footballplayer" and represented by *inst2*), which can be decomposed into two stems, "Fußball" (meaning "football" and represented by *inst3*) and "Spieler" (meaning "player" and represented by *inst8*). This is recursively continued for "Fußball" which is composed of the stems "Fuß" and "Ball" (meaning "foot" and "ball," represented by *inst5* and *inst7*).

The linguistically annotated documents are then further processed by the transformation component to populate a knowledge base of entities related to soccer (players, teams, etc.) and events (matches, goals, etc.) according to the SWIntO ontology. The mappings used here are represented in a declarative fashion specifying how the feature-based structures produced by Sprout are mapped into structures which are compatible with the underlying ontology. Further, the newly extracted information is also interpreted in the context of additional information about the match in question. This additional information is obtained by wrapping structured descriptions of football matches and also mapping them to the underlying ontology. The information obtained in this way about the match in question can then be used as contextual background with respect to which to interpret the newly extracted information. As an example, consider the linguistic annotation and ontology mapping for the following German sentence from one of the soccer game reports:

"*Guido Buchwald wurde 1990 in Italien Weltmeister*"
(Guido Buchwald became world champion in 1990 in Italy)

For this sentence, the Sprout system would, for example, produce the feature structure depicted in Fig. 16. The feature structure for `player` will be translated into the ontological structure shown in Fig. 17.

The example shows that, for the representation of the extracted information, SOBA relies on SmartSUMO to produce entities which are compliant with the ontology and, thus, can be used by other parts of the system.

As mentioned above, the mapping from feature structures to ontological structures is specified declaratively within an XML file which can be easily modified, thus allowing to enhance the mapping in a flexible manner. Fig. 18 shows an excerpt of this mapping file specifying how an instance of `NaturalPerson` with a `NaturalPersonDenomination` is created for a feature structure of type `player`. Each *simple-mapping* rule has an input and an output and generates one or more instances of a SWIntO concept or relation. In this particular case, the first rule creates an instance of the relation `dolce:firstName`, whereas the second creates an instance of the relation `dolce:lastName`. The values of the corresponding feature structures are respectively bound to the variable `VAR1` and directly used to construct the output. More complex cases can also be handled by our formalism. Instead of simple mappings, we can for example define case distinctions as well as defining more complex procedures to compose the output

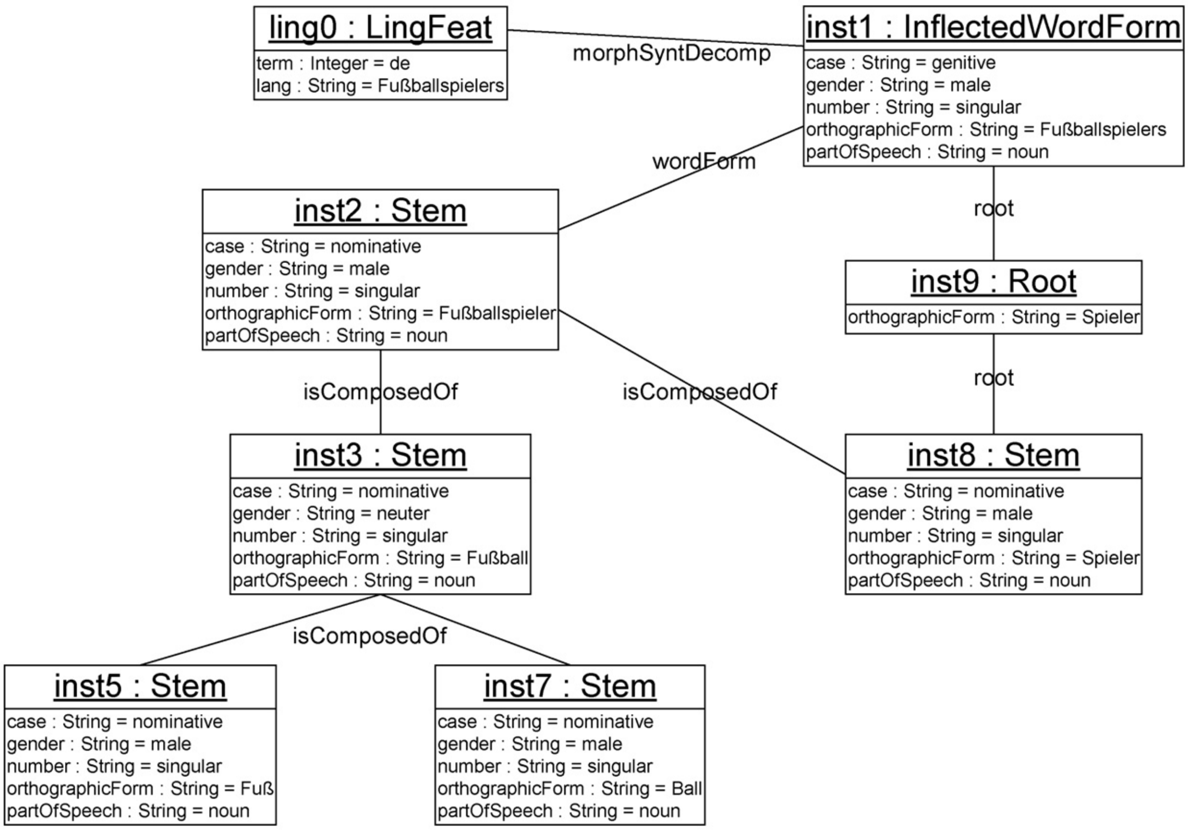

Fig. 15. Our running example of the German term “Fußballspielers” (meaning “of the football player”) depicted as UML object diagram.

```
<FS type="player_action">
   <F name="GAME_EVENT">
      <FS type="world champion"/>
   <F name="ACTION_TIME">
      <FS type="1990"/>
   <F name="ACTION_LOCATION">
      <FS type="Italy"/>
   <F name="AGENT">
      <FS type="player">
         <F name="SURNAME">
            <FS type="Buchwald"/>
         <F name="GIVEN_NAME">
            <FS type="Guido"/>
```

Fig. 16. Example Sprout output for Guido Buchwald wurde 1990 in Italien Weltmeister.

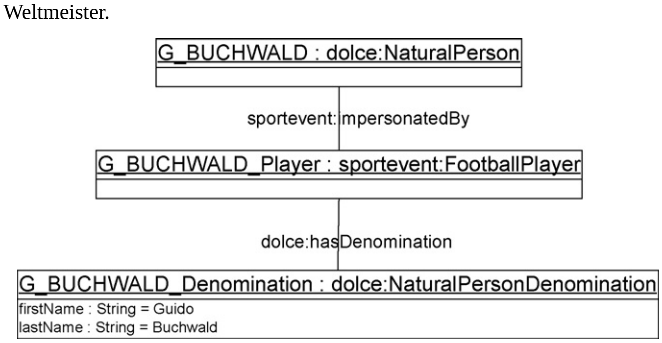

Fig. 17. Example for a feature structure represented by SWIntO depicted as UML object diagram. Concept instances are represented by objects and instantiated relations associations by object associations.

on the basis of the values of the bound variables in the input section.

SmartSUMO is also used in the linguistic analysis components when extracting temporal information. The example in Fig. 19 shows the ontological entities representing a goal achieved in the 20th minute of the game by the player denoted by the id *soba:“GRE_vs_GER_28_MAR_01_21:30_A_CHARISTEAS_PFP”* and leading to an intermediate result of ‘1:1’. Note that the timepoint of the goal is specified relatively to the start of the game at timepoint *soba:“28-MAR-01_21:30”*.

5.2. Context-dependent utterance interpretation

Context information, such as location, daytime, weather conditions, is crucial for adequate interpretation of the user’s queries. If the user does not provide this information explicitly, the SmartWeb system has to infer it. Our server-side context component is in charge of providing information about the situation in which the dialog between user and system takes place. The context information is gathered from various *context sources*, such as location (gained from a GPS receiver), time, domain, or weather-condition. Each context source is identified by a concept of the ontology and providing an instance of this concept as result.

The server-side context component is located within the dialog manager and enriches the user query by additional contextual information. It does so by either specializing general concepts or

```
<mappings>
(...)
  <type orig="player" target="dolce:natual-person-denomination>
    <link type="dolce:natural-person" method="dolce:HAS-DENOMINATION"/>
      <map>
        <simple-mapping>
          <input>
            <arg orig="GIVEN_NAME" target="VAR1"/>
          </input>
         <output method="dolce:FIRSTNAME" value="VAR1"/>
       </simple-mapping>
       <simple-mapping>
         <input>
           <arg orig="SURNAME" target="VAR1"/>
         </input>
         <output method="dolce:LASTNAME" value="VAR1"/>
       </simple-mapping>
    </map>
  </type>
(...)
</mappings>
```

Fig. 18. Example for a mapping rule from Sprout output to the ontology specified declaratively in an XML file.

placing more information-rich instances into the query representation. Specialization takes place if the user utters a hypernym instead of a more specific word. For example, he would query for the winner of "the tournament", which in a specific setting would certainly refer to "Soccer World Cup." The context component would change the `sportevent:Tournament` instance in the query to an instance of `sportevent:WorldCup`.

To model such contexts, it was very helpful to have both the *ontology design pattern* of Descriptions & Situations as described in Section 4.2 and a common *modeling basis* which facilitates interlinkage to concepts of other domain ontologies. The context component takes the following steps:

- For each instance in the interpretation of the user utterance, the associated concept from the Navigation Ontology is selected.
- The concepts representing a context are linked by relations such as `dolce:modalityTarget` or `dolce:requisiteFor`. By exploiting this, the whole context is extracted and put in an "active context" pool.
- Whenever a concept contained in a context is covered by a context source, the source will be queried. The resulting context information instance will be used to choose a more specific description.
- In another iteration over the interpretation, the instances are modified to match new and more specific context.

The following example should illustrate this: a query for the route-planning (see Fig. 9) assistance will trigger the `navigation:Locomoting` context. The containing `Environment` parameter will lead to a query of the `WeatherForecast` context source. The resulting information will be either an instance of `Rainy` or `Sunny` both being sub-concepts of `WeatherForecast`.

From this concept, a more specific description, `GoodWeatherMotorcycling` is selected and activated. Using again the linkage between DnS and the ground ontology, all instances in the utterance are compared to this new specific description. If an instance is linked to a concept in the more

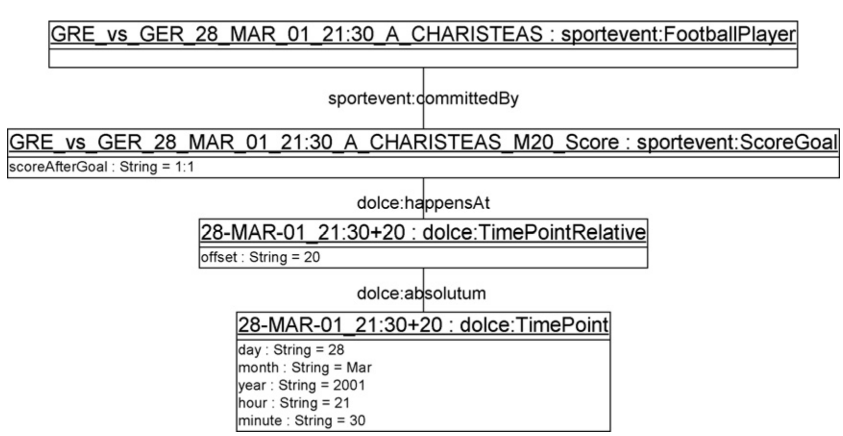

Fig. 19. Example for an extracted temporal information represented by SWIntO depicted as UML object diagram. Concept instances are represented by objects and instantiated relations associations by object associations.

general descriptions, its type will be changed to the concept linked to the new one. In this example, an instance of `navigation:Road` would become an instance of `CountryRoad` thus reflecting the fact that on a sunny day a motorcyclist would prefer using such a route rather than a highway.

In this case, the type information of the resulting context information is used. In other cases, such as time or location, the instance in the query will be replaced by the context instance.

5.3. Web services in SmartWeb

Recently, Web services are becoming a very important technological component in the application integration domain. There already exist a number of Web services for mobile applications from various content providers. Many Web services are integrated in the SmartWeb system, e.g.:

- *Navigation service*: This service supplies maps, calculates a route for a list of stations with given relations and presents traffic information.
- *Tourism information service*: The information about point of interests (such as hotels, restaurants, etc.) around a given location is provided by this service. One can also get the addresses, distances and map for an object.
- *Cinema service*: This service provides movie information (e.g., play times for the given cinema) as well as cinema information.
- *Emergency pharmacy service*: Allows searching for the emergency pharmacies on the given location and gives information on address and opening time.
- *Weather service*: The weather, ozone, UV, pollen, etc., information in Germany can be obtained by this service.
- *Webcam-finder service*: Gratuitously accessible web cameras on the Internet can be found by this service.

As an example, consider a tired tourist who is visiting Berlin by car and says to the SmartWeb system: "*I am hungry. Besides, there is almost no gas left in the car anymore. Unfortunately I don't have any cash, but only possess a visa card. SmartWeb, please help me*." In this case, the SmartWeb system interprets the user's intention and associates the expressions "*hungry*" with "*restaurant*," "*no gas*" with "*go to gas station*" and "*visa card*" with "*pay type*". Because of the word "*gas*", SmartWeb knows a car route should be searched. The current user position is determined by means of the integrated GPS devices.

The integration of Web services to answer such questions required two extensions of SmartSUMO. Without having a common *modeling basis* in the form of SmartSUMO, such extensions would have been very tedious. First, SmartSUMO had to be extended by means of modeling semantic inputs and outputs of Web services, as well as their behavior. We took parts of the Core Ontology of Services [25] and the Ontology of Plans [12] to achieve this goal. Second, some of the XML-schemata of the Web services had to be represented ontologically. Let us consider the `getRoute()` service as an example. It takes `AddressProperties[]` and `RouteInputProperties` XML-schema types as input and yields `RouteOutputProperties` as output. In this example, the `AddressProperties` XML-schema has a counterpart in the ontology, namely SmartSUMO's `Address`. This is not the case for `RouteInputProperties`. However, we have to represent this XML-schema type in the ontology in order to represent it via EMMA and let the dialog engine and context module add and set information (e.g., the speed profile which is part of `RouteInputProperties`). Hence, we straightforwardly modeled such XML-schema types as specialization of `InformationObject`.

6. Conclusion

In this paper, we presented the SmartWeb Integrated Ontology, discussed its design choices, and reported on the collaborative ontology engineering effort it required. With our ontology engineering methodology, we were able to integrate the SmartWeb domain ontologies with reasonable effort, and the interaction between ontology engineering experts and domain experts was smooth and fruitful. Thus, we feel that SmartSUMO's is a reasonable and worthwhile compromise between the cleanness of ontology design and the practical aspects of ontology use.

The experiences we have gathered via this large-scale collaborative was that *conceptual clarity* is of major importance. For example, it was quite important to avoid ambiguities for concepts, such as Hotel, which in our system can refer either to an actual building or to an answer to a user's question. The rigorous usage of *ontology design patterns* and having a common *modeling basis* helped enormously in the building of the domain ontologies and applications. Additional conventions can also help, such as for naming, which can reduce the amount of redundancies and mismatches. Nevertheless, in some cases, there is simply no substitute for personal communication, such as frequent meetings, phone calls, e-mail, and online tools such as Wikis, to resolve conflicts within a collaborative ontology engineering effort.

The contribution of our work is two-fold. On the one hand, we have defined a substantial ontology of high quality and ready for practical use. On the other hand, we have made a case in point for collaborative ontology engineering by means of foundational ontologies. We therefore consider the SWIntO ontology to be a significant step towards the practical realization and application of semantic technologies.

Acknowledgements

We would like to thank our partners in SmartWeb for useful comments on earlier versions of this article. This research was funded by the German Federal Ministry for Education and Research (BMBF) under grant number 01IMD01, *SmartWeb* (http://www.smartweb-project.org). The responsibility for this article lies with the authors.

References

[1] J. Allen, Towards a general theory of action and time, Artif. Intell. 23 (1984) 123–154.

[2] P. Buitelaar, P. Cimiano, A. Frank, Racioppa S S., SOBA: SmartWeb Ontology-Based Annotation, in: Proceedings of the Demo Session at the International Semantic Web Conference, Athens, GA, USA, 2006.

[3] P. Buitelaar, T. Declerck, A. Frank, S. Racioppa, M. Kiesel, M. Sintek, R. Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, P. Cimiano, LingInfo: design and applications of a model for the integration of linguistic information in ontologies, in: Proceedings of the OntoLex06 Workshop at LREC, Genoa, Italy, May, 2006, 2006.

[4] S. Borgo, N. Guarino, C. Masolo, A pointless theory of space based on strong connection and congruence, in: Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96), Cambridge, Massachusetts, USA, November 5–8, 1996, Morgan Kaufmann, 1996, pp. 220–229.

[5] S. Bechhofer, I. Horrocks, C. Goble, R. Stevens, OilEd: a reasonable ontology editor for the Semantic Web, in: Proceedings of the Joint German Austrian Conference on Artificial Intelligence, Lecture Notes Artif. Intell., vol. 2174, Springer, 2001, pp. 396–408.

[6] A.B. Benitez, H. Rising, C. Jorgensen, R. Leonardi, A. Bugatti, K. Hasida, R. Mehrotra, A.M. Tekalp, A. Ekin, T. Walker, Semantics of multimedia in MPEG-7, in: IEEE International Conference on Image Processing (ICIP), 2002.

[7] P. Buitelaar, M. Sintek, M. Kiesel, A lexicon model for multilingual/multimedia ontologies, in: Proceedings of the 3rd European Semantic Web Conference, ESWC2006, Budva, Montenegro, June, 2006, Springer, 2006.

[8] P. Cimiano, A. Eberhart, P. Hitzler, D. Oberle, S. Staab, R. Studer, The SmartWeb foundational ontology, SmartWeb Project Report, September, 2004.

[9] U. Callmeier, A. Eisele, U. Schäfer, M. Siegel, The DeepThought core architecture framework, in: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC), 2004, pp. 1205–1208.

[10] R. Casati, A.C. Varzi, Holes and Other Superficialities, MIT Press, 1995.

[11] W. Drozdzynski, H.-U. Krieger, J. Piskorski, U. Schäfer, F. Xu, Shallow processing with unification and typed feature structures-foundations and applications, Künstliche Intelligenz 1 (2004) 17–23.

[12] A. Gangemi, S. Borgo, C. Catenacci, J. Lehmann, Task taxonomies for knowledge content, Metokis Deliv. D07 (September) (2004).

[13] A. Gangemi, C. Catenacci, M. Battaglia, Inflammation ontology design pattern: an exercise in building a core biomedical ontology with descriptions and situations, in: D.M. Pisanelli (Ed.), Ontologies in Medicine, IOS Press, 2004, pp. 64–80.

[14] M.R. Genesereth, R.E. Fikes, Knowledge Interchange Format Version 3.0 Reference Manual. Report Logic 92-1, Logic Group, Stanford University, California, USA, 1992.

[15] A. Gangemi, F. Fisseha, J. Keizer, B. Lauser, J. Lehmann, A. Liang, I. Pettman, M. Sim, M. Taconet, An overview of the FOS Project: towards a fishery Semantic Web, Internal Project Report, ISTC-CNR, Laboratory for Applied Ontology, Rome, Italy, 2002.

[16] A. Gangemi, P. Mika, Understanding the Semantic Web through descriptions and situations, in: DOA/CoopIS/ODBASE 2003 Confederated International Conferences DOA, CoopIS and ODBASE, Proceedings, LNCS, Springer, 2003.

[17] M. Grüninger, C. Menzel, The Process Specification Language (PSL) theory and applications, AI Mag. 24 (3) (2003) 63–74.

[18] A. Gangemi, M.-T. Sagri, D. Tiscornia, A constructive framework for legal ontologies, Internal Project Report, EU 6FP METOKIS Project, Deliverable, 2004. http://metokis.salzburgresearch.at.

[19] E. Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, D. Ravichandran, Towards semantic-based answer pinpointing, in: Proceedings of Human Language Technologies Conference, San Diego, CA, March, 2001, 2001, pp. 339–345.

[20] J. Hunter, Adding multimedia to the Semantic Web—building an MPEG-7 ontology, in: Proceedings of the International Semantic Web Working Symposium (SWWS), 2001.

[21] B. Loos, R. Porzel, Towards ontology-based pragmatic analysis, in: Proceedings of DIALOR'05, Nancy, France, 2005.

[22] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K.A. Miller, Introduction to WordNet: an on-line lexical database, Int. J. Lexicogr. 3 (4) (1990) 235–244.

[23] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, L. Schneider, The WonderWeb Library of Foundational Ontologies, WonderWeb Deliverable D17, August 2002. http://wonderweb.semanticweb.org.

[24] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, Ontology library (final), WonderWeb Deliverable D18, December, 2003. http://wonderweb.semanticweb.org.

[25] P. Mika, D. Oberle, A. Gangemi, M. Sabou, Foundations for service ontologies: aligning OWL-S to DOLCE, in: The 13th International World Wide Web Conference Proceedings, ACM, May, 2004, 2004, pp. 563–572.

[26] I. Niles, A. Pease, Origins of the IEEE standard upper ontology, in: Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology, Seattle, Washington, August 6, 2001, 2001.

[27] D. Oberle, Semantic Management of Middleware The Semantic Web and Beyond, vol. I, Springer, 2006.

[28] Ch.K. Ogden, I.A. Richards, The Meaning of Meaning—A Study of the Influence of Language upon Thought and of the Science of Symbolism, Kegan Paul, Trench, Trubner & Co., London, 1923.

[29] A. Pease, Core Plan Representation, Object Model Focus Group, November, 1998.

[30] R. Porzel, I. Gurevych, Contextual coherence in natural language processing, in: P. Blackburn, C. Ghidini, R. Turner, F. Giunchiglia (Eds.), Fourth International Conference on Modeling and Using Context, Springer (LNAI 2680), Berlin, 2003.

[31] A. Pease, I. Niles, J. Li, Origins of the IEEE standard upper ontology, in: Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, Edmonton, Canada, July 28–August 1, 2002, 2002.

[32] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, Pacific Grove, CA, USA, 1995.

[33] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke, OntoEdit: collaborative ontology development for the Semantic Web, in: I. Horrocks, J.A. Hendler (Eds.), The Semantic Web—ISWC 2002, First International Semantic Web Conference, Proceedings, Lecture Notes Comput. Sci., vol. 2342, Sardinia, Italy, June 9–12, 2002, Springer, 2002.

[34] B. Smith, Mereotopology: a theory of parts and boundaries, Data Knowledge Eng. 20 (3) (1996) 287–303.

[35] D. Sonntag, Towards interaction ontologies for mobile devices accessing the Semantic Web—pattern languages for open domain information providing multimodal dialogue systems, in: Proceedings of the Workshop on Artificial Intelligence in Mobile Systems (AIMS), 2005, MobileHCI, Salzburg, 2005.

[36] J.F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, USA, 2000.

[37] A. Zipf, H. Aras, CORBA-basierter Zugriff auf SDE-Erste Implementierung der OpenGIS Simple Features Specification für CORBA in Java, in: ArcAktuell, Krankzberg, 2000.