Requirements for the Design of Electronic Dictionaries and a Proposal for their Formalisation

Dennis Spohr
Universität Stuttgart

We discuss recent analyses of the requirements for the design of electronic dictionaries, building primarily on the accounts by de Schryver (2003), Chiari (2006), Heid (2006) and Tarp (2008). These requirements suggest a richer formalization of dictionary models than is usually the case in traditional database and plain XML-based approaches, and we therefore argue in favour of a formalisation of these requirements in the framework of a strongly typed formalism. The discussion focusses on users' needs, needs of specific applications of Natural Language Processing, and multifunctionality-in the sense suggested by Gouws (2006) and Heid/Gouws (2006). We further point out the benefits of a richer formalization of dictionary models that goes beyond the traditional view on lexical resources, and strengthens our claim by providing evidence from related work on lexicon modelling in OWL DL (Burchardt et al., 2008).

1. Introduction

The structure of electronic dictionaries (EDs) is a central topic in lexicographic research, and a variety of approaches have been pursued and implemented over the past decades. Very often, however, there is an apparent gap between dictionaries serving applications of natural language processing (NLP), and those serving the needs of different types of human users. In this paper, we will have a detailed look at some of the issues that arise in the definition of models for EDs which are *multifunctional* in the sense of serving these two different purposes (cf. Heid and Gouws, 2006).

The requirements for the model of a multifunctional electronic dictionary may be subdivided into several (partially overlapping) categories: (i) *detail of descriptio*n, which has to be chosen such that the ED is capable of serving as useful input to both specialised NLP tasks and human expert users, while retaining the possibility to generate or extract less detailed descriptions from the data if appropriate; (ii) *access and retrieval* should, from a technical point of view, be scalable and performed very efficiently. From a more practical point of view the access functionality should offer means for complete exploration of all sorts of data contained in the dictionary, as well as their relations and complex combinations of both; (iii) *consistency and integrity* become increasingly important when dealing with large amounts of dictionary data, and even more if these are acquired and inserted both automatically and manually. Relevant questions in this context are e.g. which properties or relations are used to describe which kinds of dictionary items, and how it is possible to ensure that these items actually make use of only the properties they are "allowed" to; (iv) *specific users' needs* and their effects on the appearance of the ED, i.e. its general layout as well as the way in which extracted data are presented; finally, (v) *specific needs of NLP applications*—such as application programming interfaces—represent crucial requirements that have to be met in order for an ED to be truly multifunctional.

In the following, we will discuss a number of these requirements in more detail, focussing on particular aspects of linguistic description (Section 2.1) as well as formal and technical aspects (Sections 2.2 and 2.3)—both with a strong view on multifunctionality and its implications on the underlying formalism (Section 2.4). Section 3 presents the proposed formalism as well as its relation to existing formalisms for the definition of ED models, and provides evidence from recent related work in this framework. Section 4 presents specific issues in the dictionary-making process, and shows how they can be approached in the proposed formalism. The paper concludes in Section 5.

2. Requirements and their implications

The following sections explain requirements on the detail of description in the ED. The list of phenomena to be covered, as well as the required detail of their description is, of course, potentially infinite, and we therefore illustrate the requirements wrt. three different lexical phenomena which we consider particularly relevant to both human users and several NLP applications, namely valence, multi-word expressions (MWEs), and distributional as well as lexical preferences. Section 2.2 and 2.3 focus on rather technical requirements, namely access and retrieval as well as means to ensure consistency and integrity, and Section 2.4 discusses the implications of these requirements on the choice of the formalism.

2.1. *Linguistic aspects*

Valence description. Heid (2006) emphasises the importance of detailed valence descriptions with respect to both human users and NLP applications. For the production of texts, he notes that it is vital to make explicit reference to valence differences between (quasi-)synonyms like *treffen* and *begegnen* ("encounter, meet"), where the direct accusative object of *treffen* ("that which is encountered") is mapped onto the indirect dative object of *begegnen*. The same is true for machine translation, an area of NLP which relies heavily on (and greatly benefits from) detailed valence descriptions.

A number of researchers have suggested to use the three-layered approach to valence description proposed by FrameNet (Baker et al., 1998; see e.g. Atkins et al., 2003; Boas, 2005) in order to provide adequate treatment of valence phenomena in the lexicon. In this approach, the subcategorised (as well as optional) arguments of a predicate are not only assigned a *phrasal category* (as in many current valence dictionaries; see e.g. VDE) and a *grammatical function* (as in NLP lexicons such as subcategorisation lexicons of lexical functional grammar (LFG); see e.g. Butt et al. 2002), but also a *semantic role*. A valence pattern thus consists of one or several such category-function-role triples. This combination of both syntactic and semantic information in the FrameNet approach provides "an analysis of meaning far more granular than is normally possible in commercial lexicography" (Atkins et al., 2003: 340).

A further crucial point is that valence descriptions should not only be provided for verbs, but that this treatment should be extended to cover nouns (in a manner suggested e.g. by Boas 2003) and MWEs as well (cf. Heid and Gouws 2006). In line with what has been discussed for (lexical-)semantically related items, the differences in valence patterns as well as the mapping of valence arguments (i) between morphologically related items (e.g. verbs and their nominalisations), and (ii) between "collocationally related" items (e.g. nouns and their occurrences in support-verb-constructions) are of central importance. Therefore, the points that have been made so far do also apply to non-verbal lexical items and MWEs.

Multi-word expressions. Although MWEs are of utmost relevance to both NLP and language-learning tasks, adequate treatment has been largely neglected in past dictionary design. Apart from specialised collocation dictionaries which have been specifically designed to deal with MWEs (see e.g. OCDSE; DiCE), many current EDs (e.g. ELDIT) assign to them the status of usage examples in the microstructure of a lexical entry. As a result, it is in such dictionaries difficult—if not impossible—to obtain more detailed information about MWEs, such as valence descriptions (see above) or preferences (see below). Such information is, however, indispensable in order for a dictionary to be a useful lexicographic tool, especially for text production. As a consequence, MWEs should be promoted to the status of "second level treatment units" (Heid and Gouws 2006), i.e. receive a microstructural description. The resulting requirement for the model of the ED is to provide adequate representations of the phenomena associated with MWEs, and many of these have been discussed at length in the works cited above.

Preferences. Describing preference phenomena is important e.g. for the production of texts, irrespective of whether they are to be automatically generated by an NLP component or produced by a foreign language learner. An approach to detecting morphosyntactic preferences of collocations has e.g. been discussed in Evert et al. (2004). The main focus there was on extracting distributional preferences, such as quantifications about how often the base of a collocation is used in the singular or the plural (e.g. *sich Hoffnungen machen* ('to have hopes'), where the base is most frequently used in the plural; p. 907).

While it is, of course, not only MWEs which show this kind of distributional preferences wrt. morphosyntactic features, lexical items in general do further show selectional preferences in terms of valence. The focus here is not on subcategorised arguments, but rather on preferences as far as lexical fillers of specific argument slots are concerned. Heid (2006) lists *er hält X von jemandem* ('he thinks X of someone') as an example, where *X* may only be replaced by a rather restricted set of lexical fillers, such as *viel*, *nichts* or *eine Menge* ('much', 'nothing', 'a lot'; p. 77; cf. VDE). In contrast to distributional preferences, which can to some extent be extracted for morphosyntactic features on the basis of corpus text with shallow linguistic analysis, the calculation of sets of lexical fillers for subcategorised argument slots will very likely be more involving.

2.2. *Access and retrieval*

The specification of the access and retrieval functionality represents a more technical concern, and Tarp (2006) lists a number of minimal features that should be retrievable from an ED, such as idioms, lemmata, irregular forms, word class or gender. In addition to this, Chiari (2006: 144) states that combinations of such features should also be queryable, e.g. "all nouns and verbs which are rare or frequent and specific of any field except physics". Of course, such expressions can be arbitrarily extended ("...and which subcategorise prepositional phrases except ones with *auf*..."), meaning that for all items in the dictionary that are connected in some way, these connections should be explorable, possibly by different ways of access (fuzzy search, Boolean operators, etc.; cf. de Schryver 2003). Chiari's ideas are in line with what is more generally labelled as "non-standard access" in Spohr and Heid (2006: 71), i.e. "access via paths involving other properties and relations than just lemmas". In a related way, de Schryver (2003: 173) mentions "access aspects for which the outer search path (leading to a lemma sign) does not necessarily precede the inner search path (leading to data within articles)", and Atkins (1992: 521) even talked about the "iron grip of the alphabet", calling for "new methods of access". In this vein, it should in principle be possible to access the data at any arbitrary point in the model. In other words, there should not be a predefined entry or access point to the data—as is usually the case with standard lemma-based query access[1].

In contrast to the traditional view on dictionaries as lists of lexical entries—which are according to Polguère (2006: 51) simply "texts, in the most general sense"—this supports the concept of viewing a dictionary as a graph in which, among others, "implicit references, in fact, all words [...] should be hyperlinked to the relevant lemma" (Prinsloo 2005: 18), and where all nodes and edges in the graph may serve as potential access points (see also Trippel 2006). In addition to this, the entities in the dictionary should be linked to external and complementary sources of information, such as online search engines and text corpora (de Schryver 2003; Gelpí 2007; Tarp 2006). Assuming that a graph forms the basis of the dictionary model, several tools are in principle available, such as efficient storage and query engines (e.g. Tamino XML Server[2]; AllegroGraph[3] or Sesame for RDF[4] (Broekstra et al. 2002)) and visualisation software (e.g. Graphviz[5]), which can be integrated into the dictionary architecture.

[1] Whether this is desirable for all kinds of users is not the question here. We believe, however, that it is better to set the stage for "unrestricted access" to the data and later constrain it according to the type of user, than to allow only for restricted access in the first place.

[2] See *http://www.softwareag.com/Corporate/products/tamino/default.asp*.

[3] See *http://agraph.franz.com/allegrograph/*.

2.3. *Consistency and integrity*

Terminology. The final requirement for the ED model that is to be discussed here refers to rather formal aspects, namely the means that are necessary in order to ensure consistency and integrity of the ED. Although the notion of integrity is in some senses of the word subsumed by the notion of consistency, we choose to use these two terms in order to describe two separate things. For us, consistency refers to the question as to whether the underlying model of the ED is *satisfiable*, i.e. whether it is at all *possible* for lexical data to satisfy the conditions defined in the dictionary model without causing any contradictions. Integrity, on the other hand, refers to the question as to whether the data *actually* satisfy the conditions, and whether their descriptions are complete. In order to achieve this, it is necessary to be able to (i) identify and distinguish between different types of data in an ED, (ii) define different well-formedness constraints and properties for these types, (iii) restrict the set of items that can occur as values of these properties, and (iv) make sure that the data adhere to these restrictions. In the following, we will construct a few simple examples that are intended to illustrate the difference between the notions of consistency and integrity.

Example cases. For a basic example of consistency issues, consider a lexical resource that distinguishes, among others, between a type of verbal predicate that has a transitive syntactic subcategorisation frame of the form <subject, $object_{acc}$> and a type of predicate with a frame <subject, $clause_{that}$>. A conceivable formalisation of the former type would be that verbs belonging to it have as arguments only a subject and an accusative object, while a formalisation of the latter would state that verbs of this type only have a subject and a that-clause. If we now assert the verb *ankündigen* ("to announce") as belonging to both the former and the latter type, we get conflicting definitions: a verb that is of type "only subject and $object_{acc}$" can never satisfy the condition that it has "only subject and $clause_{that}$", and thus we end up with a contradiction. Although this inconsistency has been caused by a lexical item in the data set, it is without any doubt a problem of the formal definition of the model that underlies the resource, as any lexical item that belongs to both types at the same time implies the inconsistency of the resource. Therefore, the formalism has to be capable of detecting such inconsistencies, and the model needs to approach the explained phenomenon in a different way.

A very basic kind of integrity check is e.g. to ensure that the values of a part-of-speech property of lexical items are actually made up of part-of-speech tags, and not of grammatical gender, case, or misspelt variants (e.g. *n.* instead of *noun*); this should probably be possible with any "mildly" structured formalism, if the model caters for a "controlled vocabulary" of *descriptive devices* (cf. Spohr and Heid 2006) or *data categories* (ISO 12620, 1999). However, more intricate cases are conceivable, e.g. for collocations of the type V + N_{Obj} (i.e. collocations with a verbal collocator and a nominal base that is the object of the verb, such as *eine* $Rede_N$ $halten_V$ ("to $give_V$ a $speech_N$"); here, the part of speech of the base of the collocation (*Rede*) has to be N, and the collocate (*halten*) has to be a transitive verb that subcategorises an object. Such restrictions have to be formalisable and verifiable in the ED model, and traditional approaches that rely on document type definitions (DTD) or XML schemata do not have the formal means to express these kinds of restrictions.

Consistency of the underlying model is usually regarded as given—considering its sparse mention in the related literature—, and inconsistencies are assumed to occur only in the data. However, since a model is usually created at least in part manually, ensuring its consistency is an issue that needs to be addressed, especially if the complexity of the model goes beyond DTDs or XML schemata.

[4] See *http://www.w3.org/RDF/*.

[5] See *http://www.graphviz.org/*.

2.4. *Implications on the choice of the formalism*

One of the most striking requirements, which can be directly derived from the above analysis, is the fact that the underlying formalism cannot be entirely unconstrained, but rather has to be strongly typed. Hence, we do not follow very general approaches as those e.g. by Trippel (2006) and Polguère (2006), as they do not seem to provide for powerful structural means for ensuring consistency and integrity in the sense discussed above—e.g. relations with defined domain and range—and rather focus on a general and unconstrained graph structure.[6]

Instead, we propose to use a typed formalism based on the Resource Description Framework (RDF), such as RDF Schema or the Web Ontology Language[7] (OWL), which among others offer the formal devices needed to address exactly those issues mentioned. A further reason for choosing RDF is the assumption that if we attempt to define our own framework or metalanguage, it is not unlikely that we arrive at a "remodelling" of subsets of RDF—the description of items in a dictionary can be considered as a specific case of describing resources in general—except for the fact that then large parts of the existing technical infrastructure are no longer available, such as tools which interpret the vocabulary that is needed for this description. This would then mean that all but the very basic infrastructure has to be reimplemented in order to be able to interpret the "new" vocabulary, and that it is thus much more difficult to share the content. Of course, the source code can be shared easily if an appropriate platform-independent formalism like XML has been chosen, but it is by no means easy to share the interpretation of the content, such as the semantics of specific XML element tags.[8] This is not to say that all these issues dissolve once RDF is used. It rather means that using a common metalanguage that has been defined in a declarative and standard framework entails some advantages, such as the fact that—as in the case of OWL DL—the formal characteristics and complexity have been investigated extensively and are well-known, and that even at the most abstract level more than just very basic infrastructure is available, namely sophisticated editors (e.g. Protégé[9]; Knublauch et al., 2004), reasoners (e.g. RacerPro[10]), standardised interchange formats (DIG Interface[11]) and efficient querying tools (e.g. AllegroGraph; see Sections 2.2 and 4).

3. Formalisms for the representation of electronic dictionaries

In the previous section, we have argued for a formalisation of the model for an ED in a typed formalism. In the following, we will look more closely at different XML-based languages that can be used for representing ED models. XML can certainly be considered the standard framework for modelling EDs, and our focus will be on languages in the realm of the *Semantic Web*, a W3C initiative to provide design principles for modelling meaningful web content, as well as the technology to realise these principles (see Berners-Lee et al., 2001). In Section 3.2, we discuss recent attempts at dealing with the above requirements in OWL DL, as it is the most expressive of the RDF-based formalisms (cf. Görz, in prep.) and thus the most promising option for the modelling of EDs. Moreover, OWL has recently been proposed as a formalism for implementing the *Lexical Markup Framework* (LMF; Francopoulo et al. 2007).

---

[6] In part, this is motivated by their slightly different objectives, since they focus on the unification of different lexical resources, rather than the creation of new dictionaries. Therefore, they have chosen a format that allows for the "cohabitation" of different dictionaries.

[7] See *http://www.w3.org/TR/owl-ref/*.

[8] Cf. the joint project between the Universities of Tübingen (SFB 441), Hamburg (SFB 538) and Potsdam (SFB 632), which addresses, among others, the issue of sustainability of linguistic data (see e.g. Dipper et al. 2006).

[9] *http://protege.stanford.edu.*

[10] *http://agraph.franz.com/racer/*.

[11] DL Implementation Group; see *http://dl.kr.org/dig/*.

3.1. *RDF-based languages and their relation to XML*

XML, XML Schema and the concept of *namespaces* form the basic layer of the hierarchy of languages in the Semantic Web. From the viewpoint of ED design, they provide the basic means for defining custom formats for the representation of EDs (see e.g. Trippel 2006), as well as a general and convenient mechanism for combining several dictionaries and relating their entities (cf. Spohr and Heid 2006). The Resource Description Framework (RDF) is a framework which enables the expression of subject-predicate-object *triples*, which can be combined to form labelled *graphs* (see Görz, in prep.). RDF does not inherently rely on a particular formalism, and—among others—it can be expressed in XML. With the vocabulary RDFS (RDF Schema) it constitutes the next layer in the language hierarchy. RDF and RDFS allow for a hierarchical structuring of lexical knowledge by providing a mechanism for expressing *typed hierarchies*, both for classes (types) and for properties (relations). Thus, they facilitate the definition of abstractions and generalisations over lexical data (see Burchardt et al. 2008) and provide a powerful way to express underspecification in queries (cf. Spohr and Heid 2006: 68). This class and property subsumption can be considered as a very basic kind of inference, although, as Görz (in prep.) points out, RDF and RDFS have no commitment to more sophisticated inference mechanisms. In addition to this, RDFS provides the means to express domain and range restrictions on the properties defined in RDF, which is a crucial feature to realise the requirements mentioned in Section 2.3 above.

The Web Ontology Language Description Logics (OWL DL) represents the logic layer situated on top of RDF and RDFS. It extends the vocabulary with means to express Description Logic axioms (see e.g. Baader et al., 2003), which in turn enables complex inferences when connecting the model to a reasoner such as FaCT++[12] or RacerPro. In addition to this, OWL DL offers the means to formally describe properties in terms of symmetry and transitivity.

Although it is defined on top of RDFS, OWL DL cannot be seen as an extension sensu stricto, since it disallows certain language constructs which go beyond the expressivity of Description Logics in order to ensure decidability of reasoning and the complexity of DL (see Baader et al. 2003: 101 ff). Instead, OWL comes in two other sublanguages which vary according to the degree of expressivity. The least expressive, OWL Lite, is a simplification of OWL DL, while the most expressive one, OWL Full, supports all language constructs of RDFS, including e.g. the concept of *meta classes* ("classes of classes"), and is thus the only "true" extension of RDFS. However, due to its formal properties, OWL DL should be the first choice for the definition of computational dictionaries (Görz, in prep.).

3.2. *Approaching the requirements on ED models in OWL DL*

Linguistic aspects. One of the most important linguistic requirements that has been identified in Section 2.1 is the need to model detailed valence information. A recent study that has illustrated how this issue can be approached in OWL DL is the work by Burchardt et al. (2008) as part of the SALSA project (Burchardt et al. 2006), who derived an OWL DL-based computational lexicon from a corpus with syntactic and lexical-semantic annotation in the FrameNet paradigm. In their lexicon model, they make a clear distinction between a so-called *linguistic model*, representing the FrameNet and a closed set of data categories, and an *annotation model*, which distinguishes between e.g. frame annotations, role annotations and syntactic units. Both models are represented as classes which are instantiated by the annotated instances in the corpus. Through multiple instantiation, they arrive at a convenient method for modelling valence information and the syntax-semantics mapping of arguments (for more details see Burchardt et al. 2008).

For MWEs, Spohr and Heid (2006) have presented an OWL DL-based model for an ED of collocations, based on the considerations and suggestions in Heid and Gouws (2006). The major contribution of these works is that collocations and other MWEs have been assigned the status

[12] *http://owl.man.ac.uk/factplusplus/*.

of *second level treatment units*, i.e. lexical objects which can be accessed either through the microstructure of a different element or, like any single word, directly via their lemma, depending on the type of user and the according needs (cf. Tarp 2006). In addition to this, MWEs receive their own microstructural description (see Heid and Gouws 2006). The model presented in Spohr and Heid (2006) and Heid et al. (2007) also provides for a way of modelling morphosyntactic selectional preferences of collocations. These preferences have been calculated on the basis of data extracted from corpora at an earlier stage, and stored *statically* in the ED (cf. Heid et al. 2007: Section 3). In contrast to this, Burchardt et al. (2008) present a *dynamic* modelling of selectional argument preferences in that the preference values, e.g. for fillers of valence arguments, are calculated from the underlying corpus at the time of consultation. This process combines the extraction of the data from a corpus with the representation as a lexicon, which is derived from their motivation to incrementally populate the lexicon from the SALSA corpus. Their model offers a very flexible representation of lexical data and of their corpus annotations and certain tasks are delegated to the powerful query mechanism and later preprocessing stages.

Thus, the mentioned approaches show that OWL DL and the associated implementational infrastructure do not prescribe a certain way of encoding selectional preferences, but rather allow for modelling solutions with both static and dynamic representations.

Access and retrieval. Both Spohr and Heid (2006) and Burchardt et al. (2008) report very good results using the Sesame framework that was mentioned in Section 2.2 above, in combination with the query language SeRQL (Sesame RDF Query Language). One of the major advantages is that OWL DL, in combination with the mentioned query mechanism, offers a convenient way to express underspecification in queries. For example, Spohr and Heid (2006: 68) illustrate how it is possible—via hierarchical organisation of properties—to extract all semantically related items in their resource with a query using the very general relation *hasSemanticRelationTo*, while the data in the dictionary are actually linked by its more specific subproperties, such as *isSynonymOf* or *isAntonymOf*. In addition to this, the extraction of valence mappings as well as selectional preferences reported by Burchardt et al. (2008) further illustrate the power of the access and retrieval functionality in OWL DL. Finally, the inferencing capabilities even enable the extraction of bigger result sets, since further statements may be inferred, e.g., through the symmetry and transitivity of relations.

Consistency and integrity. This issue is a central topic in Burchardt et al. (2008) and also one of their strongest claims for using a typed framework for the definition of models for lexical resources. In essence, they propose a combination of general knowledge representation methods (e.g. theorem provers for axiom-based consistency checking) with Sesame's query language SeRQL. In doing so, they are able to express highly complex consistency queries involving several distinct layers, such as the formal definition of FrameNet, the frame semantic annotation scheme, as well as syntactic corpus annotations.

4. Using the Representation in the Dictionary-Making Process

4.1. *Generating dictionary entries using SPARQL*

The combination of a graph-based model in OWL with current software for efficient storage and retrieval of RDFS data offers a number of very attractive solutions for the practical realisation of multifunctional dictionaries. As a simple example, we discuss below the issue of generating dictionary entries from an underlying data collection in OWL. We will start by providing some technical background that is required in order to understand the example.

AllegroGraph is a scalable RDF graph database implementation by Franz Inc.[13] that is capable of handling billions of statements (cf. Calvanese et al. 2007: 5). It implements most of the Java interfaces in Sesame (Broekstra et al. 2002) and Jena (McBride 2001), two earlier

[13] See *http://www.franz.com*.

implementations for editing, storing and querying RDF, RDFS and OWL. By supporting the query language SPARQL (Simple Protocol and RDF Query Language), which has recently been assigned the status of a W3C recommendation (World Wide Web Consortium) and can thus be considered as standard technology in the Semantic Web, AllegroGraph offers very powerful access and retrieval functionalities that go beyond current XML-based query languages in terms of scalability and usability. In addition to this, SPARQL paves the way for a straightforward and generic method to generate dictionary entries from an OWL/RDFS repository without knowing the exact structure of the underlying source. We will illustrate this by means of a SPARQL query that extracts information about the collocation *Kritik üben* from an OWL data collection on German collocations (cf. Spohr and Heid 2006).

The interpretation of the SPARQL query in Figure 1 is that it "describes" the mentioned collocation. In particular, the query extracts a subgraph from the lexical model graph, namely all outgoing edges from the item that has the string *Kritik üben* as value of its *hasLemma* property. Apart from this, no further knowledge about the structure is required, and—in particular—no knowledge about the names of the outgoing edges.

```
DESCRIBE ?x
 WHERE {?x :hasLemma "Kritik üben"^^xsd:string}
```

Figure 1: A SPARQL DESCRIBE query used as basis for dictionary entry generation

This kind of query can be used if the *Uniform Resource Identifier* (URI; the dictionary-internal name or ID) of an entity is unknown to the user, which will typically be the case. However, if the URI of an item is known (e.g. to an internal component in the ED architecture), DESCRIBE queries can also be used directly with this name. In such cases, the query would just consist of e.g. DESCRIBE :Kritik_NN (if "Kritik_NN" is the internal name of an entity in the ED). This provides a very convenient mechanism for automatic dictionary entry generation, e.g. on the basis of programs internal to the ED (see Section 4.2 below).

The results as returned by AllegroGraph consist of sets of triples of the form *subject predicate object*, and their form can be adjusted to a custom XML format, such as the one illustrated below.

```
<triples>
     ...
   <triple>
     <subject name="#Kritik_üben"/>
     <predicate name="#hasBase"/>
     <object name="#Kritik_NN_1"/>
   </triple>
     ...
 </triples>
```

Figure 2: A custom XML format as basis for further processing by XSLT

An XML format such as this can be processed by XSLT[14] in order to produce output that can be used for further processing by NLP applications (such as LMF), to produce HTML for viewing the data in a web browser or, using e.g. XSL-FO (*XSL Formatting Objects*), to produce a PDF file for printing. A schema of the process of generating dictionary entries is presented in Figure 3.

[14] *Extensible Stylesheet Language Transformation*; see *http://www.w3.org/TR/xslt*.

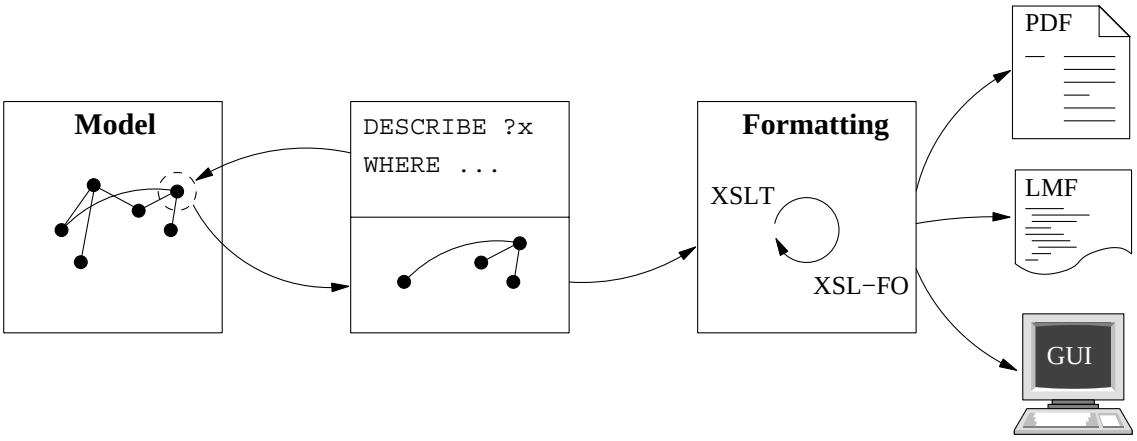

Figure 3: Schema of the process of generating dictionary entries

This simple example nicely illustrates and summarises some of the major benefits of the graph structure that was advocated in Section 2.2 above, in addition to the reasoning capabilities of OWL/RDFS: if a single-word entity such as *Kritik* ('criticism') is linked to the collocation *Kritik üben* ('to criticise') via a property like *isBaseOf*, and *isBaseOf* is the inverse of a property *hasBase*, then the statement "Kritik isBaseOf Kritik_üben" immediately triggers the inference "Kritik_üben hasBase Kritik", and thus this latter statement is also part of the graph that is returned by the DESCRIBE query in Figure 1.

4.2. *Usage scenario*

In order to give an idea of how an ED defined in OWL/RDFS works, we will outline a usage scenario that covers the various processing stages, from the dictionary consultation by a user via the presentation of the query results up to the final presentation of the dictionary entries[15]. In the course of this, we will highlight the benefits that are connected to the choice of the formalism and the computational infrastructure related to it. Richer technical detail will be given if it concerns the formalism, or if we believe that it is required for the understanding of the respective step.

Tarp (2006) lists collocational information as one of the primary user needs for text production in the mother tongue. Taking this as basis, we assume that a dictionary user is involved in such a communicative situation and consults the ED via its graphical user interface (GUI) to find out about collocations involving *Amt* ('department, post'). For the purpose at hand, it is assumed that the GUI provides for a form containing a text field for entering *Amt*, as well as a checkbox for selecting the corresponding type as "Collocation". Upon click, the form data are read by a program internal to the ED and translated into a SPARQL query (cf. Figure 4), which extracts those entities which are of type "Collocation", and for which the string value of the lemma property matches the character string *Amt*. In simpler terms, the query can be read as "select all entities of type Collocation which have a lemma that matches *Amt*". Since this type of string search represents a very common query to the ED, it is conceivable to store a very basic query skeleton into which the type ("Collocation"), the property ("hasLemma") and the desired string value (*Amt*) are inserted programmatically (see grey slots in Figure 4).

```
SELECT ?x ?y
 WHERE { ?x rdf:type :Collocation .
         ?x :hasLemma ?y
         FILTER regex(str(?y), "Amt")
       }
```

Figure 4: A SPARQL query selecting collocations whose lemma contains the string *Amt*

This query is passed on to the AllegroGraph application, which performs the look-up and

[15] The data source is the same as the one presented in Spohr and Heid (2006).

retrieval of results from the underlying data source. For results of SPARQL SELECT queries, AllegroGraph supports the *SPARQL Query Results XML Format*[16], which has also been assigned the status of a W3C recommendation. The internal representation of a fraction of the results returned by AllegroGraph is reproduced in Figure 5.

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
 <head>
  <variable name="x"/>
  <variable name="y"/>
 </head>
 <results>
  <result>
   <binding name="x">
    <uri>http://www.example.org/example.owl#Amt_scheiden_aus</uri>
   </binding>
   <binding name="y">
    <literal datatype="http://www.w3.org/2001/XMLSchema#string">
     aus Amt scheiden</literal>
   </binding>
   </result>
    ...
   <result>
    <binding name="x">
     <uri>http://www.example.org/example.owl#Amt_uebernehmen</uri>
    </binding>
    <binding name="y">
     <literal datatype="http://www.w3.org/2001/XMLSchema#string">
      Amt übernehmen</literal>
    </binding>
   </result>
   ...
 </results>
</sparql>
```

Figure 5: A fraction of the results of the query in Figure 4 in the SPARQL Query Results XML Format

This format returns the bindings of the variables *?x* (retrieving the URI of the item) and *?y* (retrieving the lemma) in the query, namely "http://www.example.org/example.owl# Amt_scheiden_aus" with lemma *aus Amt scheiden* ("to retire"), "http://www.example.org/ example.owl#Amt_uebernehmen" with lemma *Amt übernehmen* ("to accept office"), as well as several others that have been ommitted in Figure 5, such as *Amt antreten* ('to take office'), *Amt ausüben* ("to officiate") or *Amt niederlegen* ("to resign"). By use of XSLT, these data are transformed into an HTML page displayed to the user, which contains a list of the returned lemmas. The user may now be interested in details about one of the collocations shown; by clicking on the lemma, the user is provided with the indications (e.g. morphosyntactic preferences, example sentences, etc.) stored for this particular collocation. Internally, this process is the result of a combination of SPARQL and AllegroGraph: by clicking on the collocation in the list, its URI is returned, which can then be inserted into the skeleton of a DESCRIBE query for generating a dictionary entry (cf. Section 4.1)[17]. This query is passed on

[16] See *http://www.w3.org/TR/rdf-sparql-XMLres/*.

[17] In this case, one would directly use the internal name of the entity for the query. However, the user

to the AllegroGraph application, which returns the results in an XML format that can be processed in order to produce human-readable output (e.g. HTML or PDF). In other words, the combination of SPARQL and AllegroGraph offers the possibility to generate dictionary entries on the fly, at the moment of consultation, in a very efficient and straightforward way.

Although lemmas are at least part of probably almost any query result retrieved from a dictionary, this methodology can be applied to all kinds of query results, so that DESCRIBE queries could be used to generate dictionary entries also for any other data category. Although these usually serve the purpose of *describing* other items (like *hasPartOfSpeech* as a property linking lexical items to their part of speech), rather than *being described*, the inference triggered by the definition of inverse properties (see Section 4.1) creates links from the data category *back* to the lexical item (like *isPartOfSpeechOf*), which can then be retrieved using a DESCRIBE query and used in the generation of an entry.

5. Summary

We have discussed a number of requirements for EDs that have to be addressed and covered by a model for a multifunctional ED, and we have argued in favour of a typed formalism based on the Resource Description Framework (RDF) as underlying representation language. After a discussion of the core features of RDF and of the Web Ontology Language OWL, we strengthened our arguments by presenting the major advantages with respect to representing and querying lexical data, based on recent work on lexicon modelling in this framework. Moreover, we have shown how it is possible to approach specific issues related to the dictionary-making process, such as the generation of dictionary entries. Finally, we sketched a usage scenario that covers all the steps from the dictionary consultation by a user up to the final presentation of the results in the form of automatically generated dictionary entries.

neither has to know the internal name, nor is the internal name presented to him at any stage.

References

Abel, A. et al. (2002). *Elektronisches Lern(er)wörterbuch Deutsch Italienisch* [on-line]. Bolzano: EURAC. *http://www.eurac.edu/eldit* [Access date: 20 Mar. 2008].

Alonso Ramos, M. et al. (2000). *Diccionario de Colocaciones del Español* [on-line]. A Coruña: Universidade da Coruña. *http://www.dicesp.com/* [Access date: 20 Mar. 2008].

Atkins, B. T. S. (1992). "Putting lexicography on the professional map". In *Proceedings of the Vth EURALEX*. 519-526. Barcelona, Spain.

Atkins, B. T. S.; Rundell, M.; Sato, H. (2003). "The Contribution of FrameNet to Practical Lexicography". *International Journal of Lexicography* 16 (3). 333-357.

Baader F. et al. (2003). The Description Logic Handbook: Theory, Implementation and Applications. Cambridge: Cambridge University Press.

Baker, C. F.; Fillmore, C. J.; Lowe, J.B. (1998). "The Berkeley FrameNet project". In *Proceedings of the joint COLING/ACL 1998*. Montreal.

Berners-Lee, T.; Hendler, J.; Lassila, O. (2001). "The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities". *Scientific American* 284 (5). 34-43.

Boas, H. C. (2005). "Semantic Frames as Interlingual Representations for Multilingual Lexical Databases". *International Journal of Lexicography* 18 (4). 445-478.

Boas, H. U. (2003). "Frames for Nouns". In *Proceedings of the 17th ICGL*. Prague, Czech Republic.

Broekstra, J.; Kampman, A.; van Harmelen, F. (2002). "Sesame: A generic architecture for storing and querying RDF and RDF Schema". In *Proceedings of the 1st ISWC*. Sardinia, Italy.

Burchardt, A. et al. (2006). "The SALSA Corpus: a German Corpus Resource for Lexical Semantics". In *Proceedings of LREC 2006*. Genoa, Italy.

Burchardt, A. et al. (2008). "Formalising Multi-Layer Corpora in OWL DL – Lexicon Modelling, Querying and Consistency Control". In *Proceedings of the 3rd IJCNLP*. Hyderabad, India.

Butt, M. et al. (2002). "The Parallel Grammar Project". In *Proceedings of the COLING Workshop on Grammar Engineering and Evaluation.* 1-7. Taipei, Taiwan.

Calvanese, D. et al. (2007). "Software Tools for Ontology Access, Processing, and Usage". In *Deliverable TONES-D21*. *http://www.tonesproject.org/* [Access date: 20 Mar. 2008]

Chiari, I. (2006). "Performance Evaluation of Italian Electronic Dictionaries: User's Needs and Requirements". In *Proceedings of the XIIth EURALEX*. Torino, Italy.

de Schryver, G.-M. (2003). "Lexicographers' Dreams in the Electronic-Dictionary Age". *International Journal of Lexicography* 16 (2). 143-199.

Dipper, S. et al. (2006). "Sustainability of Linguistic Resources". In *Proceedings of the LREC 2006 Workshop on Merging and Layering Linguistic Information*. Genoa, Italy.

Evert, S.; Heid, U.; Spranger, K. (2004). "Identifying Morphosyntactic Preferences in Collocations". In *Proceedings of the 4th LREC*. 907-910. Lisbon, Portugal.

Francopoulo, G. et al. (2007). "Lexical Markup Framework: ISO Standard for Semantic Information in NLP Lexicons". In *Data Structures for Linguistic Resources and Applications*. Tübingen: Gunter Narr Verlag.

Gelpí, C. (2007). "Reliability of online bilingual dictionaries". In Gottlieb, H.; Mogensen, J. E. (eds.). *Dictionary Visions, Research and Practice*. Amsterdam: John Benjamins. 3-12.

Görz, G. (in prep.). "Representing Computational Dictionaries in AI-Oriented Knowledge Representation Formalisms". In Dictionaries. An International Handbook of Lexicography – Supplementary volume: New developments in lexicography, with a special focus on computational lexicography. Berlin: Mouton de Gruyter. [to appear in 2009]

Gouws, R. H. (2006). "Die zweisprachige Lexikographie Afrikaans-Deutsch—Eine metalexikographische Herausforderung". *Germanistische Linguistik* 184-185. 49-58.

Heid, U. (2006). "Valenzwörterbücher im Netz". In Steiner, P. C. et al. (eds.). *Contrastive Studies and Valency*. Frankfurt am Main: Peter Lang. 69-89.

Heid, U.; Gouws, R. H. (2006). "A Model for a Multifunctional Electronic Dictionary of Collocations". In *Proceedings of the XIIth EURALEX*. Torino, Italy.

Heid, U. et al. (2007). "Struktur und Interoperabilität lexikalischer Ressourcen am Beispiel eines elektronischen Kollokationswörterbuchs". In Rehm, G.; Witt, A.; Lemnitzer, L. (eds.). *Data Structures for Linguistic Resources and Applications*. Tübingen: Gunter Narr Verlag.

ISO 12620 (1999). *Computer Applications in Terminology—Data Categories*. Geneva: International Organization for Standardization.

Knublauch, H.; Musen, M.A.; Rector, A.L. (2004). "Editing Description Logic Ontologies with the Protégé OWL Plugin". In *Proceedings of DL 2004*. Whistler, BC.

McBride, B. (2001). "Jena: Implementing the RDF Model and Syntax Specification". In *Proceedings of the 2nd International Workshop on the Semantic Web– SemWeb'2001*. Hong Kong, China.

Mel'čuk, I.; Polguère, A. (2000). *Dictionnaire de combinatoire.* Montreal: OLST. *http://olst.ling.umontreal.ca/dicouebe/* [Access date: 20 Mar. 2008].

Oxford Collocations Dictionary for Students of English. Oxford University Press, 2002.

Polguère, A. (2006). "Structural properties of lexical systems: Monolingual and Multilingual Perspectives". In *Proceedings of the COLING/ACL Workshop on Multilingual Language Resources and Interoperability*. Sydney, Australia.

Prinsloo, D. J. (2005). "Electronic Dictionaries viewed from South Africa". *Hermes. Journal of Language and Communication Studies* 34. 11-35.

Spohr, D.; Heid, U. (2006). "Modeling Monolingual and Bilingual Collocation Dictionaries in Description Logics". In *Proceedings of the EACL Workshop on Multi-Word Expressions*. Trento, Italy.

Tarp, S. (2006). *Leksikografien i grænselandet mellem viden og ikke-viden: Generel leksikografisk teori med særlig henblik på lørnerleksikografi*. Habilitation. Department of Language and Business Communication, Aarhus School of Business.

Trippel, T. (2006). *The Lexicon Graph Model: A generic Model for multimodal lexicon development*. Saarbrücken: AQ-Verlag.

[VDE]. *A Valency Dictionary of English*. Berlin: Mouton de Gruyter, 2004.