The Connectivity of Multi-Modal Knowledge Bases*

Joachim Baumeister[1,2] and Jochen Reutelshoefer[1]

[1] denkbares GmbH, Friedrich-Bergius-Ring 15, 97076 Würzburg
[2] University of Würzburg, Am Hubland, 97074 Würzburg

Abstract. Today, large knowledge bases are developed collaboratively and in an incremental manner. Often the engineering starts with the collection and organization of informal elements, that are subsequently refined into explicit knowledge. Due to the size of knowledge bases and the collaborative setting, the analysis of the current development progress becomes an important issue. The results of that analysis usually steer the further development direction and efforts.
In this paper, we introduce a graph-based representation of general knowledge bases, containing formal and informal knowledge. We use this representation to define general and tailored connectivity measures for knowledge bases. We briefly report on the application of these measures in an industrial case study.

1 Introduction

Despite significant progress, the development of large knowledge systems is a challenging task. One of the most pressing problems is the so-called *knowledge acquisition bottleneck* stating that the success of a system mainly depends on the successful acquisition/maintenance of knowledge [13]. The bottleneck describes the following problem areas: The high development costs of knowledge acquisition and the sustainable maintenance of knowledge. Process models have been introduced to weaken the problems of the knowledge acquisition bottleneck.Furthermore, state-of-the-art knowledge acquisition tools have introduced many advances such as support for collaboration and intuitive user interfaces to minimize the efforts of knowledge acquisition, e.g., see examples in [1, 8].

Recently, the understanding of *knowledge* in a system was defined in a broader sense by the introduction of the *knowledge formalization continuum* [2]. In general, the knowledge formalization continuum is a conceptual metaphor emphasizing that the entities of a knowledge base can have different facets ranging from very informal representations (such as text and images) to very explicit representations (such as logical formulae), see Figure 1. All facets of knowledge

 In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at http://ceur-ws.org

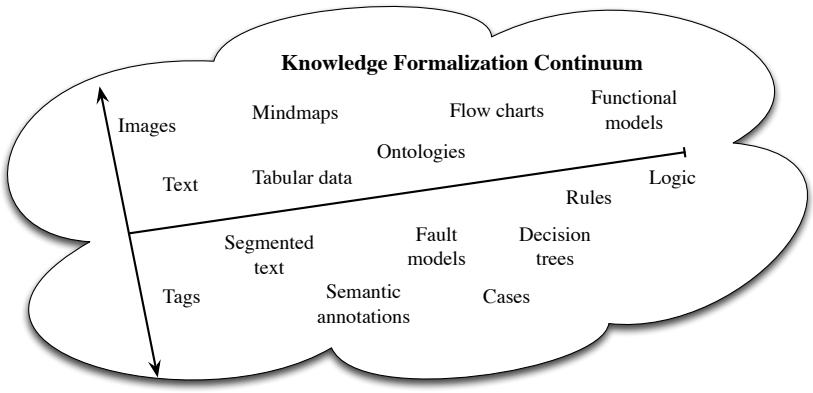

Fig. 1. The knowledge formalization continuum.

are considered as first class citizens. Thus, it is not necessary to commit to a specific knowledge facet at the beginning of a development project. Rather, it supports concentrating on the knowledge actually existing, by providing a flexible understanding of the knowledge formalization process. It is important to note that the knowledge formalization continuum is neither a physical model nor a methodology for developing knowledge bases. The concept should help domain specialists to consider even plain data, such as text and multimedia, as helpful knowledge that can be transformed incrementally to more formal representations when required. Data given by textual documents denote one of the lowest instances of formalization, represented on the left side of Figure 1. Functional models in contrast store knowledge at a very formal level, located on the right side. The term *formality* cannot be precisely defined in a general manner. In the context of our work, formal knowledge can be interpreted automatically by an inference machine. Whereas this is not possible for images today, it might be possible in a decade. A discussion of the notion of formality and the problem of its clear and useful definition (with a focus on mathemathics) is also given in [4].

The formalization of knowledge within the knowledge formalization continuum was defined as an incremental process, where knowledge is initially provided as informal chunks of documents. In iterated phases the documents are then refined into an explicit formalization that is computer-interpretable. That way, explicit resources such as input concepts, outputs, decisions, and rules are connected with the corresponding documents. Also, documents itself are connected with other documents or already existing concepts. Such a process is described for instance in [6]. Incremental knowledge formalization has some advantages:

– It is possible to fill the entire knowledge into the system very early, at least in an informal manner.

– Some areas of the knowledge are only transfered to a formalized version when beneficial. In large projects it is often reasonable to leave some parts of the knowledge base in an informal manner, see [3] for a detailed discussion.
– Informal parts of the knowledge can be used as documentation/support of the formalized counter-part.

The incremental formalization process, however, requires the regular analysis of the formalization status in order to answer the following questions:

1. How is the knowledge base generally connected by formal concepts?
2. Which parts of the knowledge base have a formalized version?
3. Which parts of the knowledge base are candidates for the next formalization increment?
4. Which formal parts of the knowledge base need further improvement?

In this paper, we propose an approach to continuously determine the connectivity of the formalization. The connectivity and especially its visualization helps to interactively answer the questions stated above. The presented approach is abstract and reusable in a way, that it can be applied to a large variety of formalization approaches, since it builds on standardized semantic technologies. In the past, the approach was applied on (scoring) rule bases, OWL ontologies, and workflow knowledge bases.

The rest of the paper is structured as follows: Section 2 introduces a graph-based notion of multi-modal knowledge bases and shows how incremental formalization is represented. The subsequent Section 3 explains the use of semantic technologies to implement the approach in a systematic manner. In Section 4 a case study is briefly described, followed by a conclusion in Section 5.

2 Connectivity Measures

2.1 Multi-Modal Knowledge

Incremental knowledge formalization is implemented on a knowledge base. In the context of our work, we define a knowledge base as an abstract graph structure.

Definition 1 (Knowledge Base as Named Graph). *Let $\mathcal{R}$ be a universal set of resources and $\mathcal{P}$ a finite set of predefined properties. A knowledge base then is a subset of all possible knowledge tuples, i.e. edges:*

$$K_{(R,P)} \subset \mathcal{R} \times \mathcal{P} \times \mathcal{R}$$

Please note, that the graph spanned by $K_{(R,P)}$ is not necessarily *connected*, i.e., some resources $r_i \in \mathcal{R}$ can be isolated, i.e., r_i has no property $p_j \in \mathcal{P}$ connecting it to another resource.

Definition 2 (Multi-Modal Knowledge Base). *Let $\mathcal{R}$ be a universal set of resources. In a multi-modal knowledge base a type from a finite set $\mathcal{L}$ of types is assigned to each resource. Further, the minimal set of properties is defined as $\mathcal{P} = \{serves, refines\}$.*

In a multi-modal knowledge base each type from the type set denotes that a resource represents a kind of knowledge resource from the knowledge formalization continuum as discussed in Section 1. For instance, $\mathcal{L} = \{M, T, D\}$ could define a type set where D represents an output value of a knowledge system, T a text paragraph, and M a multimedia object, respectively. The resources are connected by properties, for instance a *serves* property states that one resource serves as a justification for another resource. Please note, that a property can not only connect resources but also properties, e.g., the *refines* properties usually states that one property instance is refined by another property instance.

Example 1 We introduce $\{M, T, D\} \subseteq \mathcal{L}$ to be a set of types, the set of resources $R = \{<\text{T}>r_1, <\text{D}>r_2, <\text{M}>r_3\} \subseteq \mathcal{R}$, and $P = \{serves, refines\} \subseteq \mathcal{P}$. It defines that r_1 is a text paragraph, r_2 is a decision output, and r_3 is a multimedia object, such as an image for instance. The knowledge base $K_{(R,P)}$ defines the following connections:

$$K_{(R,P)} = \{serves(r_3, r_1), serves(r_1, r_2)\}$$

The property $serves(r_1, r_2)$ defines the semantic relation that the first resource r_1 fulfills a supporting/serving function for the second resource r_2. The described resources and properties are depicted in Figure 1.

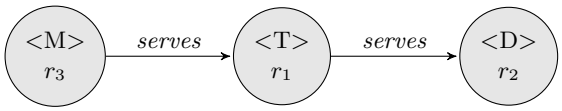

Fig. 2. A simple knowledge base $K_{(R,P)}$.

Incremental knowledge formalization represents the process of iterative extensions of a knowledge base $K \rightarrow K'$, where previously informal parts of the knowledge base K are extended by formal definitions and included in K'.

Example 2 We refer to Example 1 as the original knowledge base K. Let $R' = R \cup \{r_4, F\} \subseteq \mathcal{R}$ a set of resources and a set of properties $P' = P \cup \{refines\} \subseteq \mathcal{P}$. The type F stands for a class of formal knowledge, e.g., a rule. Then, the incremental extension $K'(R', P')$ adds a new formal input concept r_4 with the type F and the edge $serves(r_4, r_2)$, that refines the original edge $serves(r_1, r_2)$. For instance, r_4 is a rule deriving the resource r_2. The refinement relation is represented by the additional edge $refines(serves(r_4, r_2), serves(r_1, r_2))$. The incremental extension K' of the knowledge base K is depicted in Figure 2.

For knowledge based applications we distinguish two different kinds of resources (not necessarily disjoint): Resources that will be in the focus of our analysis (target resources) and resources that support the derivation of those resources (serving resources).

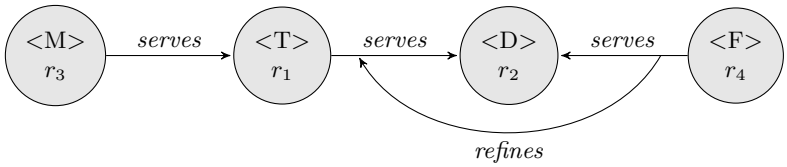

Fig. 3. The incremental extension K' of the knowledge base K.

Definition 3 (Target Output Resources and Serving Resources). *Let $\mathcal{R}$ be the universal set of resources and $\mathcal{P}$ the universal set of properties. For a knowledge base $K_{(R,P)}$ with $R \subseteq \mathcal{R}$ and $P \subseteq \mathcal{P}$ we introduce two types of special resources: We call the subset $O \subseteq R$ the* target output resources*.*

Further, the set $S \subseteq R$ of resources which are source nodes of a serves *relation are defined as the* serving resources*:*

$$S = \{\, r \mid \exists\ serves(r, x) \in KB, x \in R \ \}$$

Target output resources are used as possible outputs of the system, whereas *serving resources* support the derivation of target resources. Please note, that in larger settings also target resources can serve for the derivation of other (often more specialized) target resources. It is important to notice, that both sets—target resources and serving resources—usually grow during the knowledge formalization. For example, the refinement of one target resource can yield three more specialized target resources.

2.2 Connectivity Measures for Multi-Modal Knowledge

In the context of this paper we are interested in the connectivity of *target resources*, i.e., the use of knowledge that serves the derivation of these resources.

Simple Connectivity We define a very simple connectivity measure for general knowledge bases. Here, the connectivity of formal resources together with informal ones is calculated.

Definition 4 (Direct Connectivity). *Let $K_{(R,P)}$ be a knowledge base with $R \subseteq \mathcal{R}$ and $P \subseteq \mathcal{P}$ and a set of target resources $O \subseteq R$. Let*

$$inc(t) = \{\, p(r, t) \in K \mid r, t \in R, r \neq t \,\}$$

be the set of all incoming edges for a given resource t. For each target resource $t \in O$ the direct connectivity dcc(t) *is the number of ingoing edges in $K_{(R,P)}$:*

$$dcc(t) = |\, inc(t) \,| \quad with\ t \in O$$

The direct connectivity measure simply counts all direct links to the target resource. This measure can be refined for different types of knowledge bases: For

a decision support system, we may introduce a *static connectivity measure* that only counts edges representing explicit knowledge contained in the knowledge base. In contrary, a *dynamic connectivity measure* counts all property occurrences representing actual user input and derivation of a target resource.

Also, different subclasses of the direct connectivity measure may discriminate between the formality of the originating resource, a *formal knowledge connectivity measure* will only count edges that are describing explicit derivation knowledge. An *informal knowledge connectivity measure* will count all edges with informal knowledge as source nodes, such as text paragraphs or multimedia.

Aggregated Connectivity Often the simple counting of incoming links of a target resource does not sufficiently reflect the connectivity. When introducing strong problem-solving knowledge for the derivation of target resources, the simple connectivity is less interesting. Rather the connectivity of a target resource is reflected by its principal derivability, i.e, whether incoming edges are able to actually derive the resource or not. In this case, we need to define sub-properties of *serves*, that reflect the different possibilities of the used problem-solving knowledge. For instance, score-based knowledge requires a special *serves* property for each possible score weight. When representing Bayesian network knowledge bases, special *serves* properties need to reflect the probability.

Besides the sub-properties of serves, we also need to introduce an aggregation function *agg*, that merges all edges pointing to a target resource. It is important to note that this aggregation function *agg* needs to be tailored to the particularly knowledge representation used. We generalize the direct connectivity measure to the aggregated connectivity measure.

Definition 5 (Aggregated Connectivity). *Let $K_{(R,P)}$ be a knowledge base with $R \subseteq \mathcal{R}$ and $P \subseteq \mathcal{P}$ and a set of target resources $O \subseteq R$. For each resource $t \in O$ the* aggregated connectivity *acc is computed by the outcome of an aggregation function agg applied on all incoming properties:*

$$acc(t) = agg(inc(t)) \text{ where } t \in R$$

Please note, that the measure is not a monotonic function with respect to different formalization phases, since the set of target resources can grow during formalization.

Example 3 For the representation of a score-based knowledge base, we introduce the following three sub-properties of *serves*: $serves_1, serves_2$, and $serves_3$. Each property $serves_i$ represents a positive score weight and the respective weight can be retrieved by $w(serves_i) = i$. For the aggregation of incoming edges E we define a target resource to be connected iff the sum of weights of these properties exceeds a given *min* threshold:

$$agg_{sc}(E) = \begin{cases} 1 & : \sum_{e \in E} w(e) > min \\ 0 & : otherwise \end{cases}$$

3 Semantic Technologies and Connectivity

In the previous chapter we introduced an abstract model to jointly represent knowledge at different levels of formality. We now describe an implementation of these concepts by using semantic technologies.

```
@prefix ex: <http://example.org/ns#> .

# Triples of Example 1

ex:Target  rdf:type rdfs:Class ; rdfs:label "Target resource" .
ex:Serves  rdf:type rdfs:Class ;
           rdfs:label "serves" ;
           rdfs:comment "The subject serves/supports the object." .
ex:D       rdfs:subClassOf ex:Target ; rdfs:label "Decision" ;
           rdfs:comment "Represents the class of all target concepts." .
ex:T       rdf:type rdfs:Class ; rdfs:label "Text Paragraph" ;
           rdfs:comment "Represents the class of all text paragraphs." .
ex:M       rdf:type rdfs:Class ; rdfs:label "Multimedia" ;
           rdfs:comment "Represents the class of all multimedia resources." .
ex:r1      rdf:type ex:T ;
           rdfs:label "Lorem ipsum..." .
ex:r2      rdf:type ex:D ;
           rdfs:label "Decision 1" .
ex:r3      rdf:type ex:M ;
           rdfs:label "A picture" .
ex:p3      rdf:type ex:Serves ;
           rdf:subject ex:r1 ;
           rdf:object ex:r2 .
ex:p5      rdf:type ex:Serves ;
           rdf:subject ex:r3 ;
           rdf:object ex:r1 .

# Incremental formalization of Example 2

ex:F       rdf:type rdfs:Class ; rdfs:label "Formal" ;
           rdfs:comment "Represents formal knowledge." .
ex:r4      rdf:type ex:F ;
           rdfs:label "Rule 1" .
ex:p7      rdf:type ex:Serves ;
           rdf:subject ex:r4 ;
           rdf:object ex:r2 .
ex:refines rdf:type rdf:Property .
ex:p7      ex:refines ex:p3 .
```

Program 1: RDFS implementation of the previous examples in Turtle language.

3.1 Semantic Representation

The presented concepts can be instantly represented as RDF(S) ontology [12]. Additionally, as the de-facto standard the SKOS ontology [9] will be used to represent the hierarchical relations between resources. For the later definitions we use the Turtle language [11]. Turtle was recently published as a W3C recommendation to describe RDF data.

Within a multi-modal knowledge base we transfer all resources to RDF resources. In RDFS, we distinguish classes and instances, whereas classes are all resources that are the target of a *type* property. This convention is implemented by transferring all *type* properties to `rdf:type` properties. Analogously, we implement the *broader* property of the general definitions as the `skos:broader` property in the ontology. In Program 1 we implement the resources and properties of Example 1 as an RDFS ontology.

Please note that we added the class `Target` to represent instances of *target resources*. The class `D` represents decisions of the knowledge base and thus is a sub-class of `Target`. Also the property *serves* was not directly implemented as an RDF property but was reified as a class in order to represent refinements of *serves* relations; see for instance the implementation of relation `ex:p7`.

3.2 Querying the Connectivity

The following query shows the direct connectivity as introduced in Definition 4 as a SPARQL query [10].

```
SELECT ?broaderTarget ?targetObject ?covCount
WHERE {
  {
   SELECT ?targetObject (COUNT(?servesRel) AS ?covCount)
   WHERE {
       ?targetObject rdf:type ex:Target .
       ?servesRel    rdf:type ex:Serves ;
                     # rdf:subject/rdf:type ex:F ;
                     rdf:object ?targetObject .
   }
   GROUP BY ?targetObject
  }
  {
   SELECT ?targetObject ?broaderTarget
   WHERE {
       ?targetObject rdf:type ex:Target .
       OPTIONAL { ?targetObject skos:broader ?broaderTarget . }
   }
  }
}
```

Program 2: SPARQL query to retrieve the count of direct `serves` relations to target resources.

Besides the identifier of the target object (`targetObject`) and its number of ingoing *serves* relations (`covCount`) also the broader target resource is retrieved when available. The broader resource is required in many cases, for instance, when defining a more complex connectivity measure that also integrates the connectivities of predecessor or successor resources.

The query counts all *serves* relations independent of its degree of formality. By adding the `rdf:subject/rdf:type ex:F` to the `?servesRel` block, the query will show only *serves* relations from formal sources (the line is commented in the SPARQL query).

3.3 Visualization of Connectivity

Especially for larger knowledge bases it is essential to visualize the retrieved connectivities in order to allow for an intuitive access and overview to the connectivity state. Often target resources are organized in a hierarchical structure. Then, visualizations such as TreeMap or SunBurst are appropriate, see [7] for an evaluation work. We provide some examples for concrete visualizations in the following section.

4 Case Study

To demonstrate the ideas of this paper we report on the incremental formalization of knowledge bases in two different projects. The first project considers the development of the collaborative decision support system KnowSEC. The second case study shows the usage of a function hierarchy defined for a machine-building company.

4.1 Derivability of Decisions in KnowSEC

KnowSEC is used to support substance-related work and workflows within a unit of the Federal Environment Agency (Umweltbundesamt) by the application of knowledge based decision modules. The name KnowSEC stands for "Managing Knowledge of Substances of Ecological Concern" and the system only considers substances under REACH [5]. The multi-modal knowledge representation of the system was recently described in [3]. The KnowSEC system is an extension of the semantic wiki KnowWE [1], where informal knowledge as well as formal problem-solving knowledge and ontologies are managed. It provides plugins for automated testing and debugging knowledge bases including continuous integration.

The KnowSEC system supports the work on substances where a substance is classified according to a large number of criteria. Relevant criteria of a substance are for instance toxicity, persistence, bioaccumulation, mobility. These criteria are determined by using specialized decision modules, i.e., knowledge-based interviews that are able to automatically derive that a substance is toxic for instance. Due to the large number of criteria and its complexity of knowledge

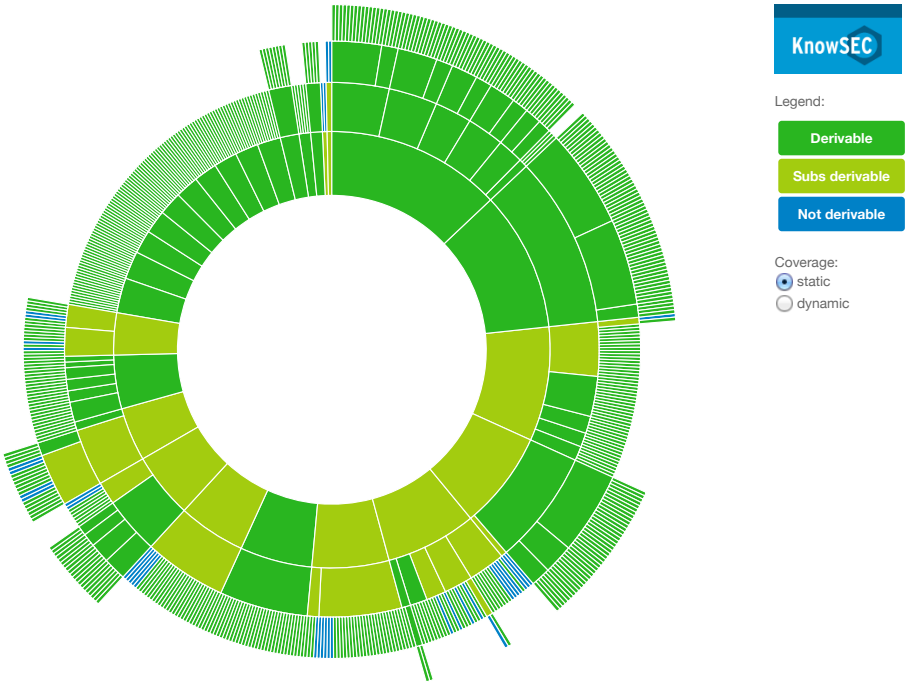

Fig. 4. Current connectivity status of the KnowSEC knowledge base.

not all criteria are covered by decision modules. Then, members of the team are writing informal justifications when applying a criteria to a substance.

The formal and informal criteria justifications are represented in an ontology as well as the possible decisions and substances. Figure 4 depicts a recent static connectivity status of the formal part of the KnowSEC knowledge base as a Sun-Burst visualization [7]. All decisions on criteria were selected as *target resources* and the direct connectivity measure was applied. We instantly see that almost all decisions of the 670 target resources or their successors are derivable, i.e., by having a *serves* relation. Also, we can point and click on segments to retrieve the name of the particular decision. Until today, different versions of the shown visualization were used during the planning.

4.2 Usage of a Function Structure

The second case study was implemented in a project with a mechanical engineering company. The developed ontology describes a function hierarchy of a large machinery, where functions/features of a broad range of machines are represented in a common hierarchical structure. During the development and right after its completion the applicability and utility of the structure was evaluated by using it in real-world use cases. In Figure 5 the hierarchical structure is shown in

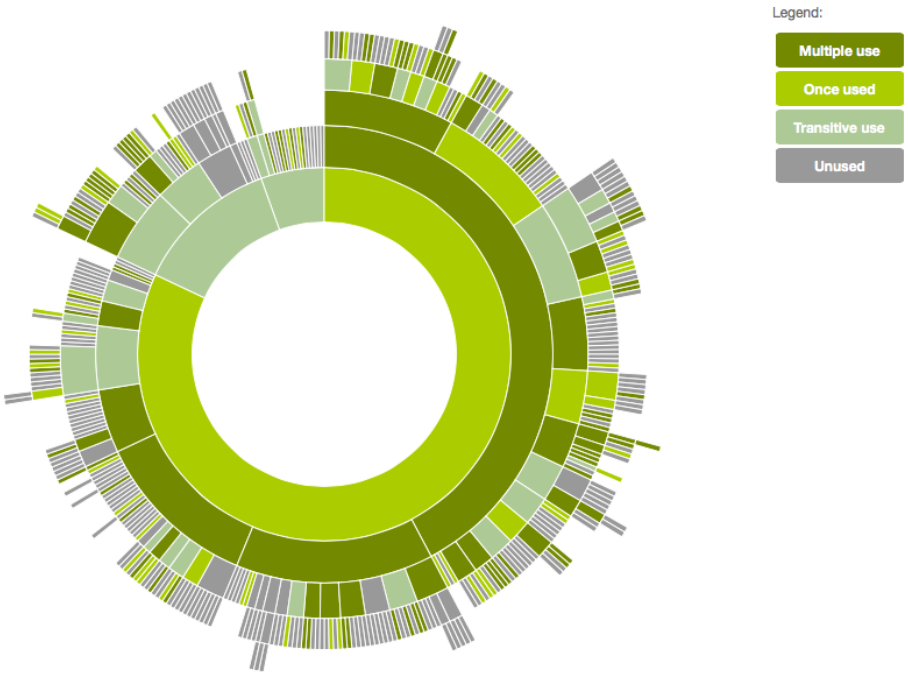

Fig. 5. Current connectivity status of an ontological symptom hierarchy.

a SubBurst visualization. Outer partitions are narrower functions, whereas inner partitions represent broader functions. The colors of the partitions represent the use of the particular function in a real-world use case. Each use was represented as a *serves* relation. Here, an aggregated connectivity measure was applied to include also functions into the analysis, that do not directly occur in the use cases but are nevertheless represented because of an occurrance of narrower functions (transitive use).

5 Conclusions

Incremental formalization can help to reduce the development risks of large knowledge bases. It proposes to initially fill the knowledge base with informal chunks of knowledge, e.g., documents and multimedia. In subsequent steps (relevant) parts of the knowledge base are incrementally formalized into a computer-interpretable format. We introduced an abstract graph-like interpretation to cope with such multi-modal knowledge representations and we showed how this interpretation can be implemented by using semantic technologies. In the introduction we posed four questions that are relevant during the formalization of knowledge bases: the connectivity, the formality, the next formalization steps, and the improvement steps. With the introduced measure the first two questions

can be answered, but it also supports the analysis of the remaining questions, e.g., unconnected and unformalized parts are typical candidates for the next formalization phase. In the best case the measures are visualized for intuitive interpretation. In two case-studies we briefly sketched their application and visualization in an industrial setting.

References

1. Baumeister, J., Reutelshoefer, J., Belli, V., Striffler, A., Hatko, R., Friedrich, M.: KnowWE - a wiki for knowledge base development. In: The 8th Workshop on Knowledge Engineering and Software Engineering (KESE2012). http://ceur-ws.org/Vol-949/kese8-05_04.pdf (2012)
2. Baumeister, J., Reutelshoefer, J., Puppe, F.: Engineering intelligent systems on the knowledge formalization continuum. International Journal of Applied Mathematics and Computer Science (AMCS) 21(1) (2011), http://ki.informatik.uni-wuerzburg.de/papers/baumeister/2011/2011-Baumeister-KFC-AMCS.pdf
3. Baumeister, J., Striffler, A., Brandt, M., Neumann, M.: Towards continuous knowledge representations in episodic and collaborative decision making. In: The 9th Workshop on Knowledge Engineering and Software Engineering (KESE2013). vol. CEUR Proceedings Vol-1070 (2013), http://ceur-ws.org/Vol-1070/kese9-03_05.pdf
4. Kohlhase, A., Kohlhase, M.: Towards a flexible notion of document context. pp. 181–188. http://kwarc.info/kohlhase/papers/sigdoc2011-flexiforms.pdf
5. Nendza, M., Müller, M., Wenzel, A.: Regulation under REACH: Identification of potential candidate chemicals based on literature, environmental monitoring, (non)european regulations and listings of substances of concern. Final report FKZ 360 12 019, Federal Environment Agency (UBA), Dessau, Germany (2009)
6. Reutelshöfer, J., Baumeister, J.: Supporting direct knowledge acquisition by customized tools: A case study in the domain of cataract surgery. In: FGWM'13: Proceedings of German Workshop of Knowledge and Experience Management (at LWA'2013) (2013)
7. Stasko, J., Catrambone, R., Guzdial, M., Mcdonald, K.: An evaluation of space-filling information visualizations for depicting hierarchical structures. Int. J. Human-Computer Studies 53, 663–694 (2000)
8. Tudorache, T., Nyulas, C.I., Noy, N.F., Musen, M.: Using semantic web in ICD-11: Three years down the road. In: The 12th International Semantic Web Conference (ISWC), In-Use Track. pp. 195–211. Springer (2013)
9. W3C: SKOS Simple Knowledge Organization System reference: http://www.w3.org/tr/skos-reference (August 2009)
10. W3C: SPARQL 1.1 recommendation: http://www.w3.org/tr/sparql11-query/ (March 2013)
11. W3C: RDF 1.1 Turtle – W3C Recommendation. http://www.w3.org/TR/turtle/ (February 2014)
12. W3C: RDF Schema 1.1 – W3C Recommendation. http://www.w3.org/TR/rdf-schema/ (February 2014)
13. Wagner, C.: Breaking the knowledge acquisition bottleneck through conversational knowledge management. Information Resources Management Journal 19(1), 70–83 (2006)