Verification of Distributed Knowledge in Semantic Knowledge Wikis

Joachim Baumeister
Institute of Computer Science,
University of Würzburg,
Am Hubland, 97074 Würzburg, Germany
joba@uni-wuerzburg.de

Grzegorz J. Nalepa
Institute of Automatics,
AGH – University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
gjn@agh.edu.pl

Abstract

Recently, the development of distributed knowledge systems has become more attractive due to the existence of new social semantic applications such as semantic knowledge wikis. User-friendly tools like wikis allow for a simple acquisition of formal knowledge, but also pose new challenges in knowledge engineering. In this paper, we reconsider classic criteria for verification in the light of a distributed knowledge base and we discuss novel anomalies that possibly occur during the collaborative development of a distributed knowledge base.

Introduction

Social semantic systems such as semantic (knowledge) wikis offer novel and fascinating ways to build intelligent systems in a collaborative way. Here, the systems are not used to build a single knowledge base but a federation of knowledge bases that are able to work together on complex problems. Semantic knowledge wikis extend traditional wikis by the representation of explicit problem-solving knowledge. In contrast to standard semantic wikis, e.g., (Krötzsch, Vrandecić, and Völkel 2006), the defined knowledge is mainly used for knowledge–intensive tasks like collaborative recommendation and classification.

The most important aspect of a wiki is the ability to change and refine its content immediately: any wiki page can be simply modified using a web browser by the mandatory edit feature of the wiki. Changes are then directly presented after saving the modifications. The distributed knowledge engineering within a wiki offers a number of benefits:

- Simple and often familiar interface of a wiki that can be used within a standard web-browser, yielding zero-installation costs for a knowledge engineer.
- Combination of explicit problem-solving knowledge with textual descriptions and multimedia information within a wiki page.
- A robust infrastructure that lets multiple developers work on a complex system knowledge base: standard wiki tools like version control, statistics, and query engines simplify the knowledge base development process.

However, the distributed development process opens new challenges in knowledge engineering when compared to the classic development of monolithic knowledge bases: The main focus of the current research seems to focus on the representation, integration and authoring of knowledge. However, with the increasing amount of knowledge contained in semantic wikis, the *quality* issues turn out to be of critical importance.

The paper discusses verification methods for distributed knowledge bases that are built in a collaborative manner. In the distributed setting the knowledge bases are able to derive solutions by their own, but also are connected by equivalent inputs and solutions, so they are able to solve more complex problems, since inputs/solutions derived by one knowledge base can be reused in another knowledge bases. In the end we arrive at a network of knowledge bases that are connected by the concepts that are used by more than one knowledge base.

In general, the verification task considers the detection of anomalies. We argue that the classic anomalies (redundancy, inconsistency, etc.) have to be rethought in the light of multiple knowledge bases and the network of knowledge bases, respectively. Additionally, a collection of new anomalies can be defined that are also important to consider in the light of distributed knowledge bases. For example, the uniform use of concepts and the level of detail of the modelled knowledge.

The rest of the paper is organized as follows: The next section introduces the concept and state-of-the-art of (semantic) knowledge wikis. Thereafter, we discuss classic criteria for verification in the light of distributed knowledge bases. However, the distributed nature of knowledge also raises new criteria; we discuss some problems in the subsequent section. In general, it is not possible to give a comprehensive list of "distributed anomalies" by now, since the implications are not yet fully understood. We conclude the paper with a discussion of the presented work and an outlook for future research.

Knowledge Wikis

First wiki systems were developed basing on the simple and well-established Content Management System (CMS) technology. As such, they did not offer any *knowledge* representation or authoring. Then, the so-called *semantic wikis*, such as the IkeWiki (Schaffert 2006), SweetWiki (Buffa et al. 2008) or Semantic MediaWiki (Krötzsch, Vrandecić, and Völkel 2006), were a next step in the direction of enriching standard wikis with the semantic technology. In such systems the standard wiki text is extended with the semantic annotations, allowing for building an ontology of the domain with which the content of the wiki is related. This extension introduces not just new content engineering possibilities, but also semantic search, navigation, and analysis of the content.

From the knowledge engineering point of view expressing semantics is not sufficient, so a knowledge-based system should provide effective knowledge representation and processing methods. In order to extend semantic wikis to knowledge-based systems the concept of *knowledge wikis* has been introduced, see (Baumeister and Puppe 2008; Reutelshoefer, Baumeister, and Puppe 2008). An example of such a system is the semantic knowledge wiki *KnowWE* (Baumeister, Reutelshoefer, and Puppe 2007). In such a system the semantic knowledge is extended with the problem-solving domain-specific knowledge. The system allows for introducing knowledge expressed with decision rules and trees related to the domain ontology.

Knowledge wikis are a new and open way to collaboratively develop and maintain knowledge bases in a distributed manner. In case on these systems the "knowledge base" is distributed over the wiki, which implies working with distributed knowledge bases. The knowledge base is also connected with text that is related to the respective wiki pages.

In principle, every knowledge base located at a particular wiki page is autonomous. It usually defines concepts for user inputs (findings) and system outputs (solutions), and interweaves these two groups of concepts by explicit knowledge, for example rules or models. When some of these concepts are also used in knowledge bases of other wiki pages, then the knowledge bases are interlinked with each other by default. The alignment is mostly done automatically, for example using simple name matching of the particular concepts, but can be improved by explicit alignment rules that match a single concept to a concept of another knowledge source. In Figure 1 the architecture for a knowledge wiki is proposed: Wiki pages are represented by so-called *knowledge services* that store the standard information (text, multimedia, etc.) of the page, but also the executable knowledge base. All known concepts (inputs and solutions) are registered in an *application ontology*, where the actually used concepts are aligned through a broker. Findings provided by the user in a problem-solving session are stored on a *blackboard*. In consequence, entered findings are propagated to each knowledge base that is aware of the corresponding concept. Furthermore, complex problem-solving structures can be implemented when outputs of one knowledge base serve as inputs of a group of other knowledge bases.

Classic Anomalies for Distributed Verification

Formal verification of knowledge-based systems is a mature field, where a number of important results have been brought up in the last decades. For some important contributions see (Chang, Combs, and Stachowitz 1990; Preece 1992; 1993; Lunardhi and Passino 1995; Lamb and Preece 1996; Vermesan 1998; Bench-Capon and et al. 1999; Vermesan and Coenen 1999; Coenen and et al. 2000). The research in verification has been especially active in the field of rule-based expert systems (Giarratano and Riley 2005). A taxonomy of formal properties for the verification of such systems has been presented in (Vermesan and Coenen 1999), with some more recent follow-ups such as (Ligęza 2006).

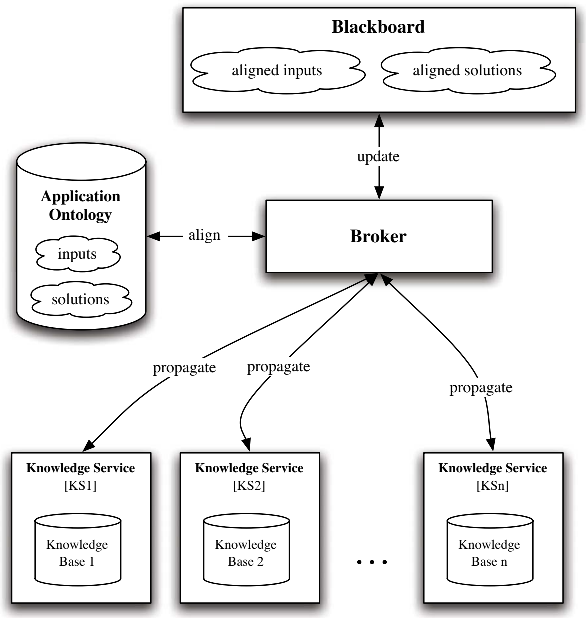

Figure 1: Architecture of a problem-solving within a knowledge wiki and its distributed knowledge bases (services).

It is worth pointing out that the actual knowledge formalization, interpretation as well as verification depend on the knowledge representation used in the intelligent systems. The most common – but not the only one – representation is based on rules, or another representation logically equivalent to rules, such as decision tables or decision trees.

According to (Vermesan and Coenen 1999; Ligęza 2006) three most important groups of properties in the verification of knowledge-based systems include: *Consistency*, *Completeness*, and *Conciseness*. In fact, these groups address a number of detailed issues, such as:

Consistency of the knowledge base means that no contradictory conclusions can be inferred from valid facts.
This issue includes:

Determinism of the system, including ambiguous rules, it may also be related to the rule inference mechanism,

Conflicts including ambivalent rules,

Logical inconsistency logical rule unsatisfiability under any interpretation.

Consistency includes the detection of ambiguous or ambivalent rules, as well as system determinism with respect to the inference strategy.

Completeness of knowledge means – in a vague sense – that no information is missing. In case of formally described systems it may mean, that the whole input space (interpreted as the Cartesian product of input attribute domains) is covered by rules. It may be interpreted, that a decision support system will find a solution for any valid question. This important property is very difficult to prove in a general case. In rule-based systems it involves the detection of missing rules.

Conciseness means, that no redundant (unnecessary) knowledge can be found in the knowledge base. This is an important issue, influencing system optimality and efficiency. Some more specific issues include:

Redundancy detection including identical, subsumed, logically equivalent, or unusable rules.

Reduction detection including rule reduction, as well as elimination of unnecessary rule attributes.

Another set of issues is related to systems that include, or are based on ontologies. In this case some ontology-specific issues need to be considered (Baumeister and Seipel 2006), including:

Circularity with respect to classes in the taxonomy.

Deficiency with respect to concept classification.

Expandability/sensitiveness that measures the costs to add new chunks of knowledge to the existing base and the impact that small changes will have on the existing knowledge base.

Starting from simple *content* management systems wikis grew as an important technology for a massive distributed *knowledge* authoring and sharing. However, to some extent they inherited some intristic limitations and drawbacks of these systems. In order to make it possible to use wikis as a *knowledge* acquisition tool, these limitations, including some wiki specific-anomalies need to be considered. Furthermore, the application of the classic verification criteria to wiki systems can than be proposed.

Wiki-specific Anomalies

Before applying the classic verification concepts to wiki systems, some important wiki-specific issues need to be identified first.

The first group of problems is related to *simple anomalies* that can be found in most of the wiki systems. The most important are:

Dead links to non-existing wiki pages.

Unreachable pages also called *orphans*, i.e., pages that are not connected with the remaining wiki content.

Another issue would include a *circular dependency* between pages, where an explanation *E1* for a problem in the page *X* is based on the explanation *E2* from page *Y*, which in fact has a link back to page *X*. However, such a circularity cannot be clearly identified in a classic wiki, where no semantic annotations are present.

The second challenge comes down to the *tacit vs. explicit* knowledge problem. Classic wikis simply store tacit knowledge embodied in the wiki text. With no formalization they provide no means for formalized verification of completeness or consistency. Only simple anomalies mentioned above can be considered. However, in the case of knowledge wikis the contents of the wiki pages can be explicitly connected to a formal model of a wiki, e.g. an ontology.

The third challenge is related to the particular *knowledge representation* used to formalize the wiki. From a formal point of view, a wiki can contain rules and facts allowing for inference, and a domain ontology allowing for classification. These concepts can only be addressed in the semantic knowledge wiki, where the knowledge representation is explicit. In these cases some representation-specific issues can be considered, e.g. ontology-specific, as mentioned above.

Another problem is related to the *interwiki connections*. In this case the possible conceptual differences between the wikis have to be considered, e.g. the relation of the domain ontology in wiki *A* and the ontology in wiki *B*. Other issues include the circular connections between the wikis, e.g., concept *C* is from page *X* in wiki *A* is explained in the page *Y* in wiki *B*, which in fact links for the explanation back to the wiki1. It is similar to the circularity problem identified before, but the interwiki nature of the connection makes it impossible to detect while analyzing only the first wiki.

For the considerations of this paper some assumptions are made. First of all, the focus is on the wikis where some kind of semantic annotations, along with some knowledge representation is used. So, only the explicit wiki knowledge should be considered, as present in the semantic knowledge wikis. Second of all, it is assumed that a wiki provides a knowledge representation that is equivalent to rules and facts, since they are a base standard for the knowledge-based systems.

So in this paper the main factors influencing wikis verification are related to:

- *knowledge distribution* (in a number of wiki pages in a single wiki) – consider what is its influence on the application of the classic knowledge-based system verification criteria to knowledge wikis with rule-based knowledge,
- *collaborative development* – consider some knowledge wikis-specific anomalies.

In the next subsection the first issue is tackled, whereas the second one is discussed in more detail in the next section of the paper.

Criteria for Distributed Verification in Knowledge Wikis

Let us consider how the three classic rule verification criteria may be applied with respect to the distributed knowledge base in a wiki system.

Wikis are composed of wiki *pages*. So a wiki can be described as a distributed knowledge-based system, where a number of knowledge bases exist. It is a distributed system, because everyone can work on his own knowledge base. Pages are usually grouped within *namespaces* related to their common semantics, which can be explicitly marked in a semantic wiki ontology. Pages in different namespaces can be interconnected, as well as pages can reference pages in other wikis (interwiki connections).

Verification Scope Considering the general knowledge wiki architecture several *verification scopes* need to be considered:

- single page scope – where the given property is analyzed only in a single wiki page, and all links are ignored.
- namespace scope – where every page in a group is considered to be a component of a single namespace-wide knowledge base, so the given formal property must hold with respect to the whole group. This means that all the links to the pages in the namespace have to be considered, whereas external links are ignored.
- wiki scope – this global wiki scope treats the whole wiki as a single knowledge base, interwiki wiki links are ignored.
- interwiki scope – in this most complex case interwiki links should be analyzed. In this paper this context is ignored, simply because current technical solutions and lack of standards make this case almost impossible to consider practically.

With respect to the above scopes the properties may be interpreted as follows.

Wiki Consistency means that no contradictory information is contained in the unit. For the practical verification the given wiki unit (page, namespace) needs to be analyzed to detect contradictory *facts* or *rules*. A simple example on the fact level could be contradictory metadata for a page:

```
<rdf:Description>
    <dc:creator>Greg</dc:creator>
    <dc:title>Tatras</dc:title>
    <dc:subject>mountains</dc:subject>
</rdf:Description>

<rdf:Description>
    <dc:creator>Greg</dc:creator>
    <dc:title>Tatras</dc:title>
    <dc:subject>tourism</dc:subject>
</rdf:Description>
```

Inconsistency is likely to appear in a distributed environment such as wiki, where a number of independent authors extend the knowledge base. It should be detected on-line, during the wiki editing session. However, it is worth noting, that considering the evolutionary nature of the wiki knowledge, such an inconsistency between two versions of a given page could be in fact a hint for knowledge refinement, so it is not obvious which of the above contradictory facts is “correct” considering the changed page contents.

Wiki Completeness means that no information with respect to the given ontology is missing in the unit. This should be considered with respect to all the pages and knowledge bases in the wiki. In a general case completeness verification is hard. It is possible to conduct such a procedure in cases where the domain of a certain property is given. This is possible in case of wikis designed according to a domain ontology. For example it would be easy to detect if it is stated that “the given page describes climate in Tatra Mountains during the whole year” and in the page descriptions for some months are missing. On the other hand, if a general mountains-related wiki is considered, then defining what mountains’ descriptions are missing might be hard and inconclusive.

Wiki Conciseness can be interpreted as a state where no redundant information is contained in the unit. In general cases it means that no identical facts or rules are inserted. In more specific cases it could also mean that new facts are more general, or that new rules subsume the older versions. For example, having two rules:

```
1) In Tatras it is cold in january.
2) In Tatras it is cold in winter.
```

The first rule could be eliminated as a redundant one given the fact that January is in winter.

In case of multi-page scope (namespace,wiki) practical implementations of the verification algorithms should consider comparing pairs of pages as units for properties verification.

A Verification Case Example

Let us consider a simple case of a community-driven wiki, describing a given geographic region, build by tourists visiting it. The idea is to gather both practical information (facts) and insights, including opinions, that could help future tourists planning the trip to the region.

In case of the *Tatra Mountains* [1] we could imagine the following namespace hierarchy (the proposal is loosely based on the corresponding Wikipedia entry referenced in the footnote):

```
- Carpathian_Mountains
-- Beskids
-- Tatras
---- Landscape
---- Climate
---- Flora
---- Fauna
---- Mountain peaks
```

We could now analyze excerpts of three different pages in the `Climate` namespace written by different tourists.

```
Page 1
 Winter lasts from november to march.
 Winters are very cold in Tatras.
 Springs start in march.
 Springs are cold.

Page 2
 Climate in Tatras is similar to Alps.
 Summer is relatively warm,
  it lasts from june to september.

Page 3
 In my opinion it is warm in Tatras.
 Summer lasts from may to september.
```

[1] See http://en.wikipedia.org/wiki/Tatras

We could observe, that in the above cases none of the pages is complete with respect of describing the whole year, additionally there is an inconsistency between page two and three in the description of summer.

Anomalies in Collaborative Development

In the previous section we categorized and discussed classic anomalies in the context of distributed knowledge bases.

Besides these formal criteria for the verification of distributed knowledge bases we can identify further aspects that especially arise due to the collaborative manner of knowledge engineering that is present (not only) in the context of knowledge wikis. In this section we sketch some aspects for the verification of collaboratively developed ontologies/knowledge bases within the *wiki scope*:

1. Heterogeneity of concepts,
2. Uselessness of knowledge,
3. Oscillating knowledge.

Each criteria is described in more detail in the following subsections.

Heterogeneity of Concepts

In the best case the distributed ontologies include concepts and knowledge at a "homogeneous" level. *Homogeneity* refers to the fact that the included knowledge bases contain definitions of concepts that are at a uniform level of detail. For example, let's consider two definitions of the concept *Temperature* defined by two ontologies:

```
Temperature (A)        Temperature (B)
- low                  - low
- normal               - normal
- increased            - high
                       - very high
```

We see that both ontologies *A* and *B* are defining a concept *Temperature* with a list of possible choice values. Whereas the first two values of each concept can be aligned very easily to each other, we can see that the remaining value(s) are more detailed for ontology *B*. Such a heterogeneous modeling of concepts become even more complex, when one concept has defined a different domain for its values, for example a concept *Temperature* expecting real values.

A heterogeneous level of detail hinders knowledge bases to exchange facts between each other thus making the distributed problem solving task impossible. For this reason, a dense network of alignments between the particular ontologies is preferred. In the literature we find mature research on the matching and alignment of ontological concepts; a recent introduction of *ontology matching* can be found in (Euzenat and Shvaiko 2007). In addition to these more or less automatic methods the (trivial) alignment of the different concepts can be enabled by *refactoring methods*, i.e., by manually renaming and fitting value ranges of specific concepts so that they naturally match with other concepts of the distributed system (Baumeister, Seipel, and Puppe 2004).

For example, Figure 2 shows the visualization of an almost perfect reuse of ontological concepts. Due to a homogeneous level of detail of the particular concepts (shown as yellow balls in the middle of the figure) more than one knowledge base (shown at the outer part of the figure) is able to use each ontology concept. Here, ontologies KB4 to KB8 are using the same 24 concepts.

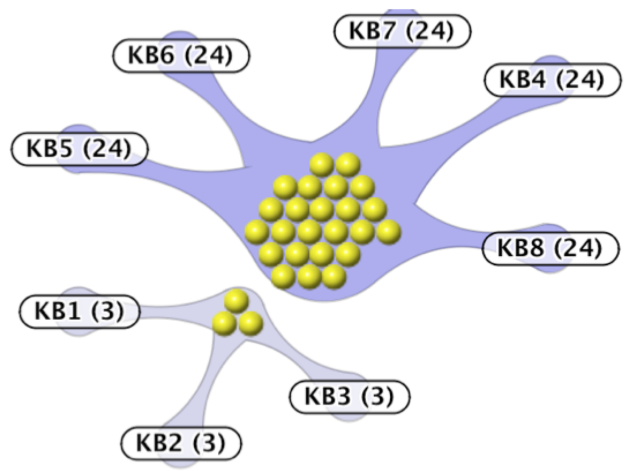

Figure 2: Almost perfect reuse of ontological concepts by the particular knowledge bases due to a homogeneous level of detail of the concepts.

Uselessness of Knowledge

The actual *usefulness* of knowledge refers to the successful use of the particular knowledge bases. Overall, the general usefulness of knowledge is difficult to measure by formal methods, but social methods can help to cope with this problem: We propose to determine the actual utility of the knowledge bases by collected user satisfaction that we retrieved by direct user feedback. For example, it is possible to provide a "feedback button" at the solution pane, where the user can vote whether the derived solution was helpful or not. Such a feedback is then back-propagated to the knowledge bases that have contributed to the derivation of the solution. With a sufficiently large number of user feedbacks it should become possible to compute a *utility value* for every knowledge base. In consequence, less useful method are promoted to the knowledge engineer for a manual inspection.

Oscillating Knowledge

Multiple and frequent edits of a knowledge base results in oscillating knowledge and point to a diverse opinion of the participating knowledge engineers. Frequent edits may refer to knowledge that is discussed a lot by the developers and for which no common sense has been found so far.

In the general wiki environment this behavior is known as *edit wars* when multiple editors of a wiki page repeatedly revert each other changes. A common measure to identify edit wars is the three-revert rule (3RR) that states that surpassing more than three edits of a page within 24 hours points to an edit war. The *oscillating knowledge* anomaly can be identified in the analogous manner. Besides the fact that such knowledge is almost not useable for problem solving, more importantly edit wars yield to personal stress that reduces motivation of the developers significantly, and that may compromise the entire project. Usually, the dispute can be damped by taking a "third opinion" into account or to open to issue to the public and request for comments.

In this section we discussed verification criteria that should be considered for collaborative knowledge bases in addition to the extension of classic verification research. The described criteria were motivated by the experience we conducted in preliminary knowledge wiki projects (Baumeister et al. 2008). However, with more experience and larger projects we are confident to build up a more comprehensive library of verification criteria.

Future Work

In the paper practical issues regarding the verification of distributed knowledge bases in wikis have been considered. Some distinct features of wikis that make knowledge verification challenging include: knowledge distribution, and the collaborative character. From these stem some new problems with verification. The paper discussed two groups of anomalies: the classic knowledge base anomalies, with respect to the distributed character of the wiki, as well as "soft" anomalies related to the collaborative nature of wikis.

Wikis are dynamic and change rapidly over time. So an important aspect considered for future work include a continuous verification of evolving wikis.

Acknowledgements The paper is partially supported by the *HeKatE* Project funded from 2007–2009 resources for science as a research project.

References

Baumeister, J., and Puppe, F. 2008. Web-based knowledge engineering using knowledge wikis. In *Proceedings of Symbiotic Relationships between Semantic Web and Knowledge Engineering (AAAI 2008 Spring Symposium)*.

Baumeister, J., and Seipel, D. 2006. Verification and refactoring of ontologies with rules. In *EKAW'06: Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management*, 82–95. Berlin, Germany: Springer.

Baumeister, J.; Reutelshoefer, J.; Haupt, F.; and Nadrowski, K. 2008. Capture and refactoring in knowledge wikis – coping with the knowledge soup. In *Proceedings of 2nd Workshop on Scientific Communities of Practice (SCOOP)*.

Baumeister, J.; Reutelshoefer, J.; and Puppe, F. 2007. KnowWE – community–based knowledge capture with knowledge wikis. In *K-CAP '07: Proceedings of the 4th International Conference on Knowledge Capture*, 189–190. New York, NY, USA: ACM.

Baumeister, J.; Seipel, D.; and Puppe, F. 2004. Refactoring methods for knowledge bases. In *EKAW'04: Engineering Knowledge in the Age of the Semantic Web: 14th International Conference, LNAI 3257*, 157–171. Berlin, Germany: Springer.

Bench-Capon, T., and et al. 1999. Validation, verification and integrity issues in expert and database systems. *Expert Update* 2(1):31–35.

Buffa, M.; Gandon, F.; Ereteo, G.; Sander, P.; and Faron, C. 2008. SweetWiki: A semantic wiki. *Web Semantics* 8(1):84–97.

Chang, C. L.; Combs, J. B.; and Stachowitz, R. A. 1990. A report on the expert systems validation associate (eva). *Expert Systems with Applications* 1(3):217–230.

Coenen, F., and et al. 2000. Validation and verification of knowledge-based systems: report on eurovav99. *The Knowledge Engineering Review* 15(2):187–196.

Euzenat, J., and Shvaiko, P. 2007. *Ontology Matching*. Berlin: Springer.

Giarratano, J. C., and Riley, G. D. 2005. *Expert Systems*. Thomson.

Krötzsch, M.; Vrandecić, D.; and Völkel, M. 2006. Semantic MediaWiki. In *ISWC'06: Proceedings of the 5th International Semantic Web Conference, LNAI 4273*, 935–942. Berlin: Springer.

Lamb, N., and Preece, A. 1996. Verification of multi-agent knowledge-based systems. *ECAI'96 Workshop on Validation, Verification and Refinement of Knowledge-Based Systems* 114–119.

Ligęza, A. 2006. *Logical Foundations for Rule-Based Systems*. Berlin, Heidelberg: Springer-Verlag.

Lunardhi, A. D., and Passino, K. M. 1995. Verification of qualitative properties of rule-based expert systems. *Applied Artificial Intelligence* 9:587–621.

Preece, A. D. 1992. Verification, validation, and test of knowledge-based systems. *AI Magazine* 13(4):77.

Preece, A. D. 1993. A new approach to detecting missing knowledge in expert system rule bases. *Int. J. of Man-Machine Studies* 38:161–181.

Reutelshoefer, J.; Baumeister, J.; and Puppe, F. 2008. Ad-hoc knowledge engineering with semantic knowledge wikis. In *SemWiki'08: Proceedings of 3rd Semantic Wiki workshop - The Wiki Way of Semantics*.

Schaffert, S. 2006. IkeWiki: A semantic wiki for collaborative knowledge management. In *STICA'06: 1st International Workshop on Semantic Technologies in Collaborative Applications*.

Vermesan, A., and Coenen, F., eds. 1999. *Validation and Verification of Knowledge Based Systems. Theory, Tools and Practice*. Boston: Kluwer Academic Publisher.

Vermesan, A. 1998. *The Handbook of Applied Expert Systems*. CRC Press. chapter Foundation and Application of Expert System Verification and Validation.