A Social Vision of Knowledge Representation and Reasoning

François Bry and Jakub Kotowski

Institute for Informatics, University of Munich
Oettingenstr. 67, 80538 München, Germany
http://pms.ifi.lmu.de

Abstract. Knowledge representation and reasoning so far have focused on the ideal ultimate goal, thus stressing logical consistency and semantic homogeneity. On the way to consistent and homogenous knowledge representation and reasoning, inconsistencies and divergent opinions often have to be dealt with. In this article, a social vision of knoweldge representation is proposed which accomodates conflicting views that possibly even result in logical inconsistencies; reasoning is used to track divergent, possibly incompatible viewpoints. This approach to knowledge repsresentation and reasoning has been developped for a social software, a social semantic wiki.

1 Introduction

Traditionally knowledge representation and reasoning focuses on processing of consistent data curated by experts in advance to conform to predfined schemes. This paper explores the needs of social semantic software, such as semantic Wikis, with respect to reasoning and knowledge representation. The phrase "Semantic Wiki" can have two basic meanings [1]: "Wiki enhanced with semantic technologies" and "Wiki for ontology engineers". This paper is focused primarily on the first sense. Several semantic Wikis were developed, e.g. [2,3,4,5], this paper presents a part of work developed in the project KiWi – Knowledge in a Wiki[1] – which develops a social semantic platform inspired by Wikis and the main application of which is a Wiki. In social software, knowledge representation is usually restricted to keyword tagging. Traditional keyword tagging can be extended in a number of ways, some of which have been investigated previously [6,7,8]. We are proposing a generalization of tagging which we call structured tagging (section 3) to provide expressive power to casual users while maintaining the simplicity of traditional tagging. Section 4 outlines how this formalism can be used together with an inconsistency-tolerant rule-based language to "emerge semantics" from user specified annotations with the help of rules. Finally, section 5 shows how a rule-language about annotations can be extended to provide further support of user-annotation and explanation by means of rule scopes and

[1] http://www.kiwi-project.eu/

tracking of origins of newly derived annotations. This work is related to CSCW[2] but is more flexible, does not rely on complex prescribed knowledge and process models. It is also closely related to work on collaborative learning with a shared knowledge artifact ("trialogical learning"), as e.g. in [9], but the focus is more specifically on supporting collaboration through automated reasoning. The concept of shared understanding is found to be very important for collaborative learning (both in synchronous learning e.g. for educational purposes and in asynchronous settings where information exchange is the main goal), see for example [10,11] for more details. In this paper we use a similar in spirit but more technical, annotation-based notion of shared understanding which is also close to a kind of social provenance information that [12] calls for. To the best of our knowledge there is currently little or no research on rule-based languages for *social* semantic environment.

2 Motivation

KiWi is an enterprise social platform that should support knowledge workers in their varied, creative work which does not adhere to predefined schemes (e.g. ontologies) and involves inadvertent transient inconsistencies. Inconsistencies are even desirable in creative process. An enterprise social semantic application should support such work and should not hinder it by enforcing unnecessary constraints. Concepts and schemes should emerge, inconsistencies should be tolerated, and users should be supported in discovering them and resolving them.

Knowledge representation and reasoning the Wiki way. Traditionally, knowledge is represented in complex formalisms by experts cooperating with domain experts. Such an approach is neither possible nor desirable in social software. Casual users should be able to semantically enrich content. The approach we suggest is to provide users with a means of creating annotations which can be made gradually more formal and precise and which can accomodate inconsistencies and incompatible viewpoints.

Emergent semantics: from informal to formal knowledge representation. Traditional approaches to knowledge representation impose a rigid prior structure on information. In contrast, collaborative tagging leads to emergence of folksonomies[3] advantages of which over predefined taxonomies have been discussed before. We argue that, given proper tools, user collaboration can gradually lead to semantically rich structures. We suggest that the right tools are structured tags and inconsistency-tolerant rule-based reasoning which can accomodate both formal (e.g. RDF/S) and informal (e.g. simple tags and structured tags) annotations.

[2] Computer-supported collaborative learning

[3] We use the word *folksonomy* to mean a user-generated taxonomy. Folksonomy is implicitly present in the bag of all tags but can also be extracted and represented the way traditional taxonomies are represented.

Negative information In real life, people are used to working with negative information. We suggest *negative tagging* as a way to express negative information in KiWi. For example someone can tag a page as "*–risky*" to express that in his or her opinion that the project described on the page is not risky. A manager could be interested in finding all projects that are not risky and have not been reviewed yet. This query refers to explicitly assigned negative tag "*–risky*" (explicit negation) and to a missing tag "*reviewed*" (negation as failure). We argue that users will benefit from both kinds of negations as they increase expressivity of the knowledge representation formalism.

3 Conceptual Model

This section shortly reviews the basic concepts a user will interact with within the KiWi system, focus is on concepts most relevant to reasoning. It is based on work published in [13], refer there for a detailed discussion of the conceptual model.

Content Items. Content items, the primary unit of information in the KiWi wiki, are composable, non-overlapping documents. Every content item has a unique URI and can be addressed and accessed individually. As a consequence, there is by default no inherent distinction between Wiki pages and content items.

Annotations. Annotations are meta-data that can be attached to content items and which convey information about their meaning or properties. Annotations can be assigned by the user, either manually or via rules. Content items and tag assignments also have system meta-data such as their creation date and their author(s). There are three kinds of annotations: simple tags, structured tags, and RDF triples whereas simple tags can be seen as a special case of structured tags. Tags allow to express knowledge informally, that is, without having to use a pre-defined vocabulary. RDF triples are used for formal knowledge representation, possibly using a pre-defined vocabulary. Structured tags can serve as a step in-between: they are structured but not yet formal.

RDF. The KiWi system chooses the RDF/S language [14,15] to specify its ontologies [16] because it is in many ways simpler than OWL [17] and yet it is sufficient for most purposes in KiWi. Simple and structured tags are represented in RDF using a predefined vocabulary in order to make them accessible to semantic querying and reasoning.

Tags. For the purpose of this paper we identify a tag with the content item that (optionally) describes it. Therefore each tag has a unique URI and can be referred to by its name (label) which can be disambiguated by cooperation of the user with the system.

Tagging. The assignment of a tag, a tagging, means the association of a content item with a tag and a user (who assigned the tag). The assignment additionally includes maintenance information. A tagging is thus a tuple consisting of a tag, a user, the tagged page, and maintenance information needed for a processing of taggings such as information about the origin of the tagging, the date when the tagging was created, a marker which allows to distinguish between explicit, derived, and system taggings. Tagging can be either positive or negative. This is distinguished by a *polarity property*. Negative taggings are displayed with a minus ("–") sign directly in front of their tag label.

Negative taggings. Although negative tagging could be seen as classical negation, it in fact is only a very weak form of classical negation because only pure taggings can be negated, not general formulae (or sets of taggings), and the only way to interpret this kind of negation is by introducing a rule which says that from tagging "t" and tagging "–t" a contradiction symbol should be derived.

Structured tags. Structured tags can be used as an intermediate step between simple tags and formal RDF annotations. Two basic operations lie at the core of structured tagging: grouping and characterization. Grouping, denoted "()", allows to relate several (complex or simple) tags using the grouping operator. The group can then be used for annotation. Example: a Wiki page describes a meeting that took place in Warwick, UK and involved a New York customer. Using atomic tags, this page can be tagged as "*Warwick*", "*New York*", "*UK*" leaving an observer in doubts whether "*Warwick*" refers to the city in UK or to a town near New York. Grouping can be used in this case to make the tagging more precise: "*(Warwick, UK), New York*". Note that, if Warwick was already defined in RDF, it could be disambiguated using its URI. Structured tags help when concepts are undefined and thus have no URI yet.

Characterization enables the classification or, in a sense, naming of a tag. The characterization operator, denoted ":", can be used to make the tagging even more precise. For example, if we wanted to tag the meeting Wiki page with a geo-location of the city Warwick, we could tag it as "*(52.272135, -1.595764)*" using the grouping operator. This, however, would not be sufficient as the group is unordered. Therefore we could use the characterization operator to specify which number refers to latitude and which to longitude: "*(lat:52.272135, lon:-1.595764)*" and later perhaps specify that the whole group refers to a geo-location: "*geo:(lat:52.272135, lon:-1.595764)*".

An important feature of structured tagging is that it allows users to work with concepts which are not fully clear. Users can begin with only a set of tags, later group them and only then realize that they, in fact, are describing a specific concept. In contrast, a formalism such as RDF/S makes its users to think in terms of concepts or classes from the very beginning and therefore constrains them in cases when the concepts are not yet clear.

The meaning of a structured tagging rests, for the most part, on the user who specified it. Structured tags do not impose strict rules on their use or purpose, they only allow users to introduce structure into taggings. It can be seen as a Wiki-like approach to annotation which enables a gradual, bottom-up refinement process during which meaning emerges as the user's work and understanding develop.

Minimal rules are, however, necessary in order to ensure a flexible, useful order and to avoid unfruitful chaos. Grouping can be applied on positive and negative simple tags and groups. Groups are unordered, cannot contain two identical members (e.g. "*(Bob, Bob, Anna)*" is not allowed), can be nested, are equal to a tag when the tag is the only one in the group, i.e. "*(Anna)*" is equal to "*Anna*". Characterization can be used on positive and negative atomic tags and groups. Characterization is not commutative, i.e. "*geo:x*" is not the same as "*x:geo*". Structured tags have to be syntactically correct, e.g. "*Bob:190cm,90kg*" is not a valid structured tag.

The above described way of structuring geo-location is only one of several. Other may include: "*geo:(x:y):(1,23:2,34)*" or "*geo:1,23:2,34*", and many more. This heterogeneity is an advantage. It provides users with freedom and it can be beneficial for different communities to encode similar kinds of information differently as the precise meaning of a similar concept may differ. Consider e.g. geo-tagging using different coordinate systems in different communities – different geo-tagging encoding would facilitate automatic translation of structured tags to formal annotations because it would allow to distinguish the two concepts structurally.

4 A Social Approach to Knowledge Representation – Emergent Semantics

Structured tags allow users to enrich ordinary tags by structuring them in order to clarify their context or qualify the thought statement that the tag is supposed to express. These structural hints can then be used to translate structured tags to formal RDF annotations via rules that specify their mapping to a predefined ontology. The advantage is that the simplicity of tagging is preserved: users can first tag and make the tagging more precise only later and eventually map it to a formal annotation. The approach could thus be summarized as "Tag, Think, Qualify, Map" – a cycle consisting of four steps. Let us take a look at an example.

Consider an enterprise Wiki in which each employee has a profile page. Alice, a junior assistant from the HR department of a company, has just hired a new employee, Bob. She creates a new page in the Wiki for Bob and decides to tag it with a few pieces of information about him. She assigns two tags to the page: "*manager*" and "*programmer*" because it is what came to her mind first. Later she realizes that the tagging can be confusing. Therefore she qualifies the tags to make it apparent that Bob is a manager with a programming background: "*position:manager*", "*experience:programmer*". Claire, a more experienced colleague of Alice, notices Alice's tags and lets her know about a new Wiki feature that their IT team is about to introduce. It is an org-chart widget that, given a name of an employee, displays the name of his or her manager and employees which report to him or her. The widget would be a nice addition to profile pages, so Alice and Claire talk to a contact from the IT department to ask how to include it. It turns out to be rather easy. The Wiki will automatically display the org-chart on any user page. User page is a page with type ucont:Employee defined in the company's UCONT[4] RDF/S ontology. Therefore all it takes to use the widget is to create a new rule in the Wiki:

[4] UCONT stands for "use-case ontology."

$position : manager \rightarrow rdf : type\ ucont : Employee$. It translates each structured tag "*position:manager*" to a formal RDF annotation "*rdf:type ucont:Employee*". Alice and Claire don not have to learn anything about ontologies and RDF/S and they still can indirectly enrich the Wiki content with formal annotations. Maybe, in the future, another widget will be able to summarize skills of an employee in which case Alice and Claire will benefit from their distinguishing between e.g. "*experience:programmer*" and "*position:programmer*".

This admittedly simplified example indicates how knowledge can emerge from initial tagging and user collaboration inside and outside a Wiki environment by taking advantage of structured tags and rules.[5] Rules could of course be more sophisticated. In the above scenario users could add the rule: "$type : \$TypeName \rightarrow ucont : \$TypeName$" and use simple structured tags of the form "type:typeName" to specify RDF/S types from the company's predefined ontology. Using similar rules, user taggings could be enriched with information from e.g. a SKOS[6] thesaurus.

5 A Social Approach to Automated Reasoning

Reasoning suitable for social software such as Wikis should be in line with their main defining traits: simplicity and collaboration. This section proposes an approach to automated reasoning which supports users collaboration and facilitates user understanding of reasoning.

5.1 Shared Understanding

During collaborative work it is important that every member knows what other members think about a subject in question. In a Wiki-like environment, collaboration takes place in and around content items. Users' understanding of a content item in KiWi can be expressed via annotations. Therefore the set of annotations of an item could be seen as the *shared understanding* of the item of all users who annotated it. Inconsistencies found within a shared understanding may be indicative of disagreements or misunderstandings of the item between the users. Such inconsistencies can be important for the future success and efficiency of the collaborative effort. The reasoning we propose is able to work in presence of inconsistencies and to point them out to users. Note that inconsistencies can emerge through constraint rules, are treated as special symbols that can be derived but cannot be matched in rule bodies and thus the classical principle "ex falso quodlibet" (anything can be inferred from a contradiction) is avoided (see [18] for details).

[5] Note that structured tags are only an unrestrictive formalism. The actual implementation is likely to provide a more convenient UI metaphor for them than textual which should also allow for easy continuous development of existing annotations.

[6] SKOS stands for Simple Knowledge Organization System – a data model to support the use of thesauri and other classification schemes, see http://www.w3.org/2004/02/skos/

Often it is also important to know who worked on what topic, who approved which content item or who agreed with something. The reasoning system we propose will keep track of how which annotation originated – depending on which users and which content-items. Therefore the system is capable of answering questions such as: "What are the inconsistencies stemming from annotations originating in content items X, Y, and Z?" and "What follows from information entered by Alice, Bob, and Claire?"

Authorship of derived annotations. The author of a derived annotation is defined as the author of the rule by which the annotation was derived. An annotation may be derived based on annotations by multiple users – this kind of information is called the user-origin of the annotation. Similarly, the content-item origin of an annotation contains information about which content-items the derived annotation depends on.

5.2 Rule Scope

In professional context the quality and source of information is often very important. For collaborative work it means that it can be effective only when people know with whom they collaborate. Consider the example from the previous section again. It could be the case that only the marketing department is allowed to decide which bugs must be fixed. Rule R_2 should therefore apply only to tags by users who work in the marketing department; the rule's *scope* should be limited only to marketing users.

Limiting the rule scope with respect to users is only one of several options. Rule scope restricts a rule to only certain data. Data in a Wiki are usually created by *users* and divided into *pages*. Therefore, content items are another candidate for a rule scope.

User scope and content item scope. Users and content items are the most important candidates for rule scope in KiWi. After all, almost everything in KiWi is represented by a content item, including users. However, a distinction has to be made between users and content items with respect to scope. Restricting a rule to a set of users understood as a set of content items would restrict them to the actual content items and not to the tags entered by these users.

Other notions of scope. There could be other candidates for a rule scope, notably time – to restrict a rule to only a certain period of time. There is, however, a significant difference between time as scope and users or content items as scope. Users and content items are a kind of provenance data of derived annotations meaning that they change according to who and where created the annotations used in the rule antecedent. Time, in contrast, does not propagate via rules. This means that while a rule with a time scope can easily be rewritten to an equivalent rule without time scope, it is not as simple in the case of rules with a user or content item scope. Such rules still could possibly be rewritten but they would necessarily have to "walk the tag dependency tree" and gather all the (user and content item) information on the way. This would have to be done before each rule application for each tag. So, at the very least, it would be inefficient. Therefore a further support from the rule language and reasoning is necessary for efficient evaluation of rules with user and/or content item scopes. The focus in the following is therefore on user and content item scopes as they are more than a mere syntactic sugar.

Extensional vs. intensional scope. So far, it was silently assumed that the user and content item scope is defined extensionally, i.e. that it consist of an explicit list of users and an explicit list of content items. The groups could also be defined *intensionally*: a rule could be meant to be applied only to tags by managers who have worked for the company for at least three years. The KiWi system will support extensional scopes. Intensional groups remain as a possible future extension because they can increase the amount of interdependencies between rules and in a language with negation as failure lead even to reasoning loops. In the following text, the words *group* and *scope* refer to the extensional meaning unless stated otherwise.

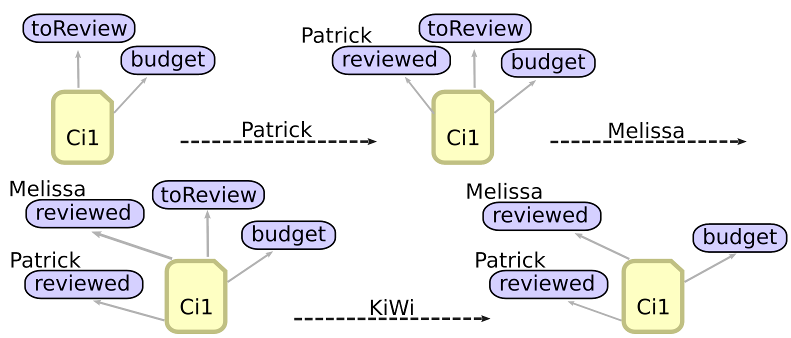

Fig. 1. A scenario illustrating how rule scope affects negation as failure. The applied rule is "(*not reviewed* → *toReview*)*WithScopeMarketing*", where $Marketing = \{Melissa, John\}$.

Rule scope and negation as failure. Rule scope limits rule application to only certain data. Negation as failure in a rule is also subject to this limitation. See Figure 1. If a rule, such as rule R_1, with negation as failure has scope consisting of two users, Melissa and John, then a negated tag in the rule body will be satisfied even if there exists this tag but is assigned by someone else than Melissa or John. In the scenario in Figure 1, Patrick tags content item "Ci1" as "*reviewed*" which does not lead to removal of the tag "*toReview*" because Patrick is not in the scope of rule R_1. The "*toReview*" tag is removed only after Melissa tags the content item as "*reviewed*". Rule scope in combination with negation as failure therefore leads to a concept similar to Axel Polleres's scoped negation as failure [19].

5.3 Computing Shared Understanding

Shared understanding of a group of users of a group of content items is the set of all tags of these content items assigned by these users. Knowing what a shared understanding looks like may help users to assess the influence a group of users or a group of content items has on the overall state of knowledge. Thus, shared understandings can play the role of additional explanation of the system behaviour. This section outlines how shared understandings can be computed in an efficient fashion. For space reasons, only user-origin of annotations is discussed in detail, for content-item-origins analogical analysis can be made.

Definition 1 *Author of an explicit annotation is the user who assigned the annotation. Author of an implicit annotation is the user who created the rule by which the annotation was derived.*

User-origin of an annotation consists of all the users based on whose annotations the annotation was derived. As there can be multiple ways to derive a certain annotation, the user-origin can consist of multiple sets of users.

Definition 2 *User-origin of an annotation is a set of sets of users. User-origin of an explicit annotation t, denoted $UO(t)$, is a set containing a singleton set containing the author of the annotation. User-origin of an implicit annotation t, $UO(t)$, derived via rule $B \rightarrow t$, where $t_1, ..., t_n$ are all annotations in the rule body B, is*

$$UO(t) = \{\bigcup_{i=1}^{n} s_i | s_1 \in UO(t_1) \text{ and } \ldots \text{ and } s_n \in UO(t_n)\}.$$

Consider the following example: $UO(bug) = \{\{Melissa\}\}, UO(-processed) = \{\{John\}\}$. The user-origin of the "*todo*" tag derived by rule $bug \wedge -processed \rightarrow todo$ is then $UO(todo) = \{\{Melissa, John\}\}$. If, in addition, it was possible to derive the tags "*bug*" and "*-processed*" by other rules and taggings by John's and Melissa's colleagues Alice and Bob and the user-origins were $UO(bug) = \{\{Melissa\}, \{Alice, Bob\}\}$ and $UO(-processed) = \{\{John\}, \{Alice\}\}$ then the origin of "*todo*" would be

$$UO(todo) = \{\{Mel., John\}, \{Mel., Alice\}, \{Alice, Bob, John\}, \{Alice, Bob\}\}.$$

$UO(t)$ intuitively consists of such "groups" of users where each group "agrees on / supports" the annotation t. Or in other words: the distributed knowledge[7] of each group of users from $UO(t)$ implies the annotation t.

Definition 3 *Shared understanding of a set of users U of a set of content items C, designated $SU(U, C)$, is*

$$SU(U, C) = \{ t \mid t \in A(C) \text{ and } (\exists s \in UO(t)) s \subseteq U\},$$

[7] in the modal logic sense

where $A(C)$ is a set of annotations assigned to content items C. It is a set of annotations where the user-origin set of each annotation contains a set which is a subset of U.

User-scope of a rule is a set of users. Each rule has a user-scope. The default user-scope is the set of all users.

Looking at the above example (by Definition 2), it is easy to see that tracking origins can lead to a combinatorial explosion if all possible combinations users and all possible combination of content items are tracked. For n users there are $n!$ possible user-origin sets. On the one hand, users can ask a question about any of these user-origin sets, on the other hand, it is likely that they are interested mainly in the sets that correspond for example to teams of their company. A team is basically a group of employees, in our case that would be a group of users. It is likely that teams would be defined as groups of users in an enterprise Wiki. Therefore, it is reasonable to focus on tracking origin-sets with respect to predefined groups of users and content items. This way the combinatorial explosion can be avoided while providing the same functionality to users.

User-origin sets with groups. When tracking user-origin sets using groups, the way of computing the origin sets is basically the same, only each origin set is replaced by the smallest predefined "group" which subsumes the origin set[8]. The user-origin set is then a set of predefined groups. For this to work correctly and as expected, a number of groups (from the full lattice of groups) has to be added. First, a group containing all users has to be added to the predefined groups so as to ensure that there always is a smallest encompassing group. Also, a group for each single user needs to be added as it is natural to require that user-scopes are possible to define for single users too.

With groups, the shared understanding of a set of users U can be computed the same way as in Definition 3 as long as the set of users U is one of the predefined groups. If the set of users is not one of the predefined groups then:

$$SU(U,C) = \bigcup_{i \in I} SU(G_i, C) \restriction (G_i \cap U)\,, \tag{1}$$

where $|G_i \cap U| \geq 1$ for all $i \in I$, G_i are all the predefined groups, and $A \restriction G$ filters out all annotations from the set of annotations A the user origin-set of which does not contain a set which is a subset of G, and $A \restriction \emptyset = \emptyset$. Equation 1 is correct under the assumption that each rule has one of the predefined groups as a user-scope (because then if U has a subset U' which is not subset of any G_i then there are no derivations dependent on tags by users U'). A consequence of this assumption is that the administrator of the system can influence what gets precomputed by creating predefined groups. The assumption could be generalized to allow the scope to be a set of groups. This would allow the user to restrict a rule to a combination of teams (resp. teams and employees) instead of restricting it to a single group.

[8] This is correct if each rule scope is one of the predefined groups.

Content-item origins with groups. Content-item origins can be adapted in a similar way to allow to work with groups of content-items. A group of content items could be imagined as a working space of a group of users which is separate from other content items with respect to annotations and other users. In this sense, each group of content items could be seen as a "knowledge space" - a closed group of content items "generating knowledge" which is self-contained: depends only on these and no other content items. If the KiWi system included a mechanism for dividing pages into strictly separate groups (e.g. with different access rights, etc.), the mechanism of content-items groups as rule scope would allow to ensure this separation at the level of reasoning too.

5.4 Explanation

Explanation is a desirable feature of a user-friendly system enhanced with reasoning. The approach to explanation and user-friendliness in KiWi is manifold and inherent in the general approach. Zacharias [20] points out four principles for building tool support for the creation of rule bases: interactivity, visibility (users should be informed about possible rule interactions), declarativity (all aspects or rule-base development should be declarative, i.e. including "debugging"), modularization (prevention of unintended interaction by providing means to modularize the rule base). The KiWi reasoning and explanation system tries to build on these principles. For example rule scope helps to modularize the rule base by limiting possible rule interactions. There are plans to support true modularization – of the rule base and of the knowledge base. Modularization of knowledge base is one step towards providing higher efficiency and therefore responsiveness and interactivity of the system as a set of rules will operate only on a specific, relevant part of the whole knowledge base (determined by the author of the rules). Of course, this is complemented by an explanation system which presents an interactive, pre-processed derivation tree of any derived annotation or inconsistency. In addition to a derivation tree, origins of derived facts are displayed too so that users can spot possible culprits of conflicts and inconsistencies more easily.

6 Conclusion

In this paper we presented a rule-based language about annotations suitable for a social semantic environment such as semantic Wikis. The approach stresses the importance of user-friendliness and lenience of both the formalism and reasoning by tolerating inconsistencies, allowing rules about not fully specified semi-formal annotations (structured tags) and we also show how simple explanations can be enhanced and enriched.

Acknowledgements. The research leading to these results is part of the project "KiWi - Knowledge in a Wiki" and has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211932.

References

1. Krötzsch, M., Schaffert, S., Vrandecic, D.: Reasoning in semantic wikis. In: Reasoning Web Summer School 2007. (2007) 310–329
2. Auer, S., Dietzold, S., Riechert, T.: Ontowiki - a tool for social, semantic collaboration. In: International Semantic Web Conference. (2006) 736–749
3. Oren, E.: Semperwiki: a semantic personal wiki. In: Proceedings of 1st Workshop on The Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure, Galway, Ireland (2005)
4. Schaffert, S., Westenthaler, R., Gruber, A.: Ikewiki: A user-friendly semantic wiki. In: 3rd European Semantic Web Conference (ESWC06), Budva, Montenegro (2006)
5. Popitsch, N., Schandl, B., Amiri, A., Leitich, S., Jochum, W.: Ylvi - multimediaizing the semantic wiki. In: Proceedings of the 1st Workshop "SemWiki2006 - From Wiki to Semantics", Budva, Montenegro (2006)
6. Bar-Ilan, J., Shoham, S., Idan, A., Miller, Y., Shachak, A.: Structured vs. unstructured tagging a case study. In: Proceedings of the WWW 2006 Collaborative Web Tagging Workshop. (2006)
7. Sereno, B., Shum, B., Motta, E.: Formalization, User Strategy and Interaction Design: Users' Behaviour with Discourse Tagging Semantics. In: Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge, WWW 2007. (2007)
8. Yang, J., Matsuo, Y., Ishizuka, M.: Triple tagging: Toward bridging folksonomy and semantic web. ISWC07 (2007) 14
9. Tzitzikas, Y., Christophides, V., Flouris, G., Kotzinos, D., Plexousakis, D., Spyratos, N.: Emergent Knowledge Artifacts for Supporting Trialogical E-Learning. InternatIonal Journal of Web-based LearnIng and TeachIng TechnologIes (3) (2007) 19–41
10. Mulder, I., Swaak, J., Kessels, J.: Assessing group learning and shared understanding in technology-mediated interaction. Educational Technology & Society (1) (2002) 35–47
11. Hinds, P., Weisband, S.: 2. In: Virtual teams that work: Creating conditions for virtual team effectiveness. Jossey-Bass (2003)
12. Harth, A., Polleres, A., Decker, S.: Towards a social provenance model for the Web. In: Workshop on Principles of Provenance (PrOPr), Edinburgh, Scotland. (2007)
13. Bry, F., Eckert, M., Kotowski, J., Weiand, K.: What the user interacts with: Reflections on conceptual models for sematic wikis. In: Proceedings of the Fourth Semantic Wiki Workshop (SemWiki 2009), ESWC 2009. (2009)
14. Manola, F., Miller, E.: Rdf primer (2004)
15. Brickley, D., Guha, R.: Rdf vocabulary description language 1.0: Rdf schema (2004)
16. Gruber, T., et al.: A translation approach to portable ontology specifications. Knowledge acquisition (1993) 199–199
17. McGuinness, D.L., van Harmelen, F.: Owl web ontology language overview (2004)
18. Bry, F., Kotowski, J.: Towards reasoning and explanations for social tagging. Proc. of ExaCt2008 - ECAI2008 Workshop on Explanation-aware Computing. Patras, Greece (2008)
19. Polleres, A., Feier, C., Harth, A.: Rules with contextually scoped negation. In: ESWC. (2006) 332–347
20. Zacharias, V.: Tackling the Debugging Challenge of Rule Based Systems. In: Enterprise Information Systems: 10th International Conference, Iceis 2008, Barcelona, Spain, June 12-16, 2008, Revised Selected Papers, Springer (2009) 144