The Semantics of Collaborative Tagging System

Milorad Tošić and Valentina Milićević

University of Niš,
Faculty of Electronic Engineering,
Intelligent Information Systems Lab – InfosysLab,
Ul. Aleksandra Medvedeva 14, PoBox 73, 18000 Nis,
Serbia and Montenegro
mbtosic@yahoo.com , valentina@elfak.ni.ac.yu

Abstract. In this paper, we adopt a system-oriented approach to the collaborative tagging and define it as a set of interactions in the system of Web resources. First, the system of Web resources is modeled as a set of interacting agents and collaborative tagging is represented as concurrent initiation of interactions between agents in the system. Also, we define concept of knowledge for individual agents. Later we use concepts of interaction and knowledge to give definition of a Link. Then, for a given Universal Set of Resources, we introduce Tag Cloud System (TCS) and definition of (possibly fuzzy) collections of resources. Finally, we introduce concept of Class, based on projection of collections of resources in the TCS, to lay down some of the groundwork towards TCS-based type system.

Introduction

In its essence, Web is all about resource locators (URLs), resource identifiers (URIs) and resource names (URNs) [1] distilling resource as one of the most fundamental concepts of the Web. Until recently, Web was considered only within its original hyper-text framework: web pages are network retrievable text documents, easy to render for human visual consumption, that may contain hyper-links to other web pages. However, massive adoption of the Internet and particularly broadband "last mile", have changed the very nature of the Web that has now been declared "Web as platform". So, Web is not anymore for human eyes only but it is also Web of data. Two different technological and philosophical methodologies are the most visible now days: Semantic Web [2] and Web 2.0 [3]. In spite of the impression that some tension exists between these two communities, we consider Semantic Web and Web 2.0 as two sides of the same coin addressing the same gap between how current technology is applied and the new opportunities. The difference is in the philosophy – general vs. simple: Semantic Web is based on a firm theoretical background and pursues a rigorous, generic top-down approach. In the same time, Web 2.0 is extremely flexible, based on ultimately simple, easy to use and easy to understand stuff, adopts bottom-up approach and worships architecture of participation (services get better as the number of users increases), collective intelligence and long tail model [4].

In [5], authors are concerned with knowledge acquisition for software development, and accordingly they define tagging as chinking and indexing knowledge acquisition dialogue using structures that are relevant to software development. However, the collaborative tagging is more traditionally considered within a framework of strategies that can be used in order to classify and organize content [4,6,7]. The classification strategies are characterized by several distinguishing attributes: If each item may be associated to exactly one category then the strategy is *exclusive*. If each category belongs to a more general one until the root of the tree is attained then the strategy is *hierarchical*. Strategy that is exclusive and hierarchical is called *taxonomy*. One of the typical examples of the taxonomy is the hierarchical directory set up by Yahoo Inc. as an impressive attempt to grow a kind of universal Web taxonomy. Tagging system is a non-hierarchical and non-exclusive strategy where each item is being assigned a list of keywords, called tags. All the tags are at the same level. The tagging systems are further classified by means of who defines the set of words or phrases that may be used as tags and who assigns tags to items. The set of tags may be defined by a central authority, such as editor or a librarian, or may include any word composed of letters. Tags may be assigned to items by the same central authority or by community. For example, in the ACM Computing Classification System [8], the central authority defines the set of keywords that may be used to classify a paper while author of the paper assigns selected keywords to the paper[1]. *Collaborative Tagging (Folksonomy)* is an ad hoc classification scheme that Web users invent as they surf to categorize the data they find online. Consequently, it is *anarchic* (the choice for the keywords are not restrained by any central authority but may be any string of alphanumeric characters) and *democratic* (the tagging is performed by a large ensamble of people, and not by a central one) [7]. Social software – software that enables users to share information and collaborate online – makes these tags available to other users, who can than take advantage of all this tagging to search for the information they need [4]. This approach has become increasingly popular, and some Web sites (call them Web 2.0 or not?) maintain ***tag cloud***, a list of all tags used on the domain usually with a visual indication of individual tag's popularity. The *collaborative filtering* is a democratic method of classification that does not require tags to be words only. The collaborative filtering exploits user access patterns to link items to people who use it [7].

We can identify three orthogonal dimensions of the concept of *scripting language*: 1) *Language* characteristics that identify a programming language as a scripting language (weak typing or even no typing at all, reflection and introspection, etc.); 2) *System* that is programmed by the scripting language (in the case of OS shell scripting languages the system is set of OS commands, while in the case of MSVisualBasic the system is composed of a set of registered ActiveX and/or COM components); and 3) *Application* under development. In this paper, we are focused on theoretical foundations for the second aspect, i.e. we envision Tag Cloud System as a system that will

[1] To the best of our knowledge, there is no tagging system like ACM CCS where set of possible keywords is defined by a central authority while readers assign the list of tags (or at least are allowed to edit it) to the paper instead of author of the paper.

be programmed by future semantic scripting languages in order to develop whole new set of global scalable applications for the “Web as a platform”. Introduction of semantics into the traditional scripting brings in two additional levels of freedom: 1) Using existing scripting languages to develop semantic applications (e.g. JavaScript programs a client side while Ruby on Rails, PHP and scripting language for programming plug-ins in Wiki are on a server side), and 2) Using “semantic scripting language” to develop not exclusively semantic applications, but also traditional applications, such as CMS for example.

In this paper, we consider collaborative tagging in a way that addresses the problem on today’s Web of bridging the gap between wide adoptability, easy to use, and simplicity from one side, and ability to address problems in a general way by adoption of the formal foundation. First, the system of Web resources is modeled as a set of interacting agents. We adopt a system-oriented approach to the collaborative tagging and define it as a set of interactions in the system of Web resources. Also, we define a concept of knowledge for individual agents based on their local state. Later we use concepts of interaction and knowledge to give definition of a Link. In the third section, for a given Universal Set of Resources, we introduce Tag Cloud System (TCS) and definition of (possibly fuzzy) collections of resources. In the fourth section, we introduce concept of Class, based on projection of collections of resources in the TCS. In this way, we lay down some of the groundwork towards TCS-based type system. Finally, we discuss a few pointers for future work and give some concluding remarks.

Semantics of the Concept of Resource

Debate over the concept of resource

In the early days of the Web, semantics of the resource concept has been much less important comparing to application and adoption aspects of the concept. As a natural consequence, the concept of resource was traditionally comprehended as *a network ‘retrievable’ entity*. However, mass adoption of the Web has resulted in completely new understanding of the value of the Web. For example, Semantic Web is one of the most promising candidate prospects. For the Semantic Web, understanding of the concept of resource is of the paramount importance because transferring data is not enough any more: Now, we have the need to communicate knowledge. To do so, we have to move up the ladders of abstraction, adopt a higher meta level as an operational level, and manipulate with knowledge and interaction instead of data and communication. Having that in mind, it is somewhat surprising that there is still an ongoing debate over definition of the resource in the literature as well as in the community [1,7,10,11,12].

Although there is a stated definition of a resource in the URI RFC, it is in many respects vague: “*A resource can be anything that has identity*. Familiar examples include an electronic document, an image, a service (e.g., ‘today's weather report for Los Angeles’), and a collection of other resources. *Not all resources are network ‘retrievable’*; e.g., human beings, corporations, and bound books in a library can also be considered resources. *The resource is the conceptual mapping to an entity or set of entities*, not necessarily the entity which corresponds to that mapping at any particular instance in time. Thus, a resource can remain constant even when its content – the entities to which it currently corresponds – changes over time, provided that the conceptual mapping is not changed in the process." [13].

The ongoing debate about the difficult problem of semantics of the concept of resource is very important, probably should receive a stronger support from at least one official standardization organization, and involves very diversified and heterogeneous scientific disciplines. In this paper, we do not want to get involved into the debate being aware of possible inconsistency in the rest of the paper. Instead, we give the following statement, based on [13,14], and consider it as correct enough for the purpose of the paper:

Resource is a generic term for anything in the universe of discourse that has identity.

Though, having in mind very limited implementation value of the statement [1,7,10,11,12], we allow further refinements in the rest of the paper on as needed bases. Comparing our previous statement about resource to the definition given by WordNet [15] that a resource is "a source of aid or support that may be drawn upon when needed (*the local library is a valuable resource)*" we may say that, by our statement, knowledge about identity of anything in the universe of discourse has a value on its own.

Multi-agent interpretation

In our system, there are two first-class meta-classes of objects: 1) *Resource* and 2) *Link*. All further constructs are built upon these two meta-classes of objects. As a modeling foundation for the definitions of the resource and link concepts, we adopt an approach that follows distributed knowledge theory developed by Joseph Halpern[2], particularly work on knowledge-based protocols [16]. In the following, we introduce basic system modeling concepts using body of work from [16] as a foundation. However, we use the concepts introduced that way in substantially new manner such that they provide foundation for presenting some of our original ideas, particularly ones related to the concept of interaction.

Let us given set of entities $AG=\{ag_i |\ i=1,2,\ldots,n\}$, called *agents*, such that each agent in the set carries certain amount of its own local information. The agent may change its local information and any change of the local information is observable by the agent. The local information is also called the agent's *local state*, $s(ag_i)$.

[2] See http://www.cs.cornell.edu/Info/People/halpern/abstract.html for the complete list of his work.

Definition 1: The set of agents, $AG=\{ag_i| \ i=1,2,\ldots,n\}$, that may ever exist in any system under consideration is called an *universal space of resources U*, also referred to as the *universe of discourse*.

We consider the given set of agents $AG=\{ag_i| \ i=1,2,\ldots,n\}$, to be closed: There is one agent in the set, called *environment*, that models state and interactions that are out of scope of the modeled system. In other words, there is no agent outside the set of agents that any agent from the set interacts with, ever: $(\forall ag_i \in AG)\ ag_i \bullet ag_j \Rightarrow ag_j \in AG$ .Note that the set of agents is not considered closed on its own sake. Instead, it is closed with respect to the modeled system. In other words, the set of agents represents our knowledge about the modeled system. Also, it represents structure that we use to reason about the system.

If agent ag_i may change local state of some other agent ag_j, or if agent ag_j may observe a (certain) change in the state of the agent ag_i, then we say that agents ag_i and ag_j are *interacting*, and such their setting is called *interaction,* $\rho_{ij}: ag_i \bullet ag_j$. Let us denote the set of all interactions between agents from the set AG as $R_{AG}=\{\rho_{ij}| \ ag_i , ag_j. \in AG\}$. We also say that an *interaction protocol* is initiated between two interacting agents. Any single agent may be involved into zero, one or more interactions. Part of the local state $s(ag_i)$ that may be changed by the agent ag_j or that is observed by the agent ag_j within the interaction ρ_{ij}, is called *projection of the local state on the interaction*, and denoted as $s(ag_i)\downarrow \rho_{ij}$. Union of projections of local state $s(ag_i)$ on all existing interactions, $s^O(ag_i) = \cup s(ag_i)\downarrow \rho$ *for all* $\rho \in R_{AG}$, is called *observable part of the local state*. *Local state of the interaction* ρ_{ij} , denoted as $s(\rho_{ij})$, is defined as a union of projection of the local state on the interaction for each agent involved into the interaction, $s(\rho_{ij})= s(ag_i)\downarrow \rho_{ij} \cup s(ag_j)\downarrow \rho_{ij}$. An agent ag_i is called *passive (active) with respect to interaction* ρ_{ij}, if it cannot (may) change local state of the interaction. A *passive agent* is an agent that is passive with respect to all existing interactions. An agent that is not passive is called *active agent*. Observable part of local state of a passive agent can be changed only as a consequence of interaction with an active agent. However, we say nothing about non-observable part of the local state – meaning that an agent may change non-observable part of its state and still be considered as a passive agent.

For example, intended semantics of the interaction may be illustrated by means of a shared variable between two concurrent threads, where threads represent agents, *ThrdA* and *ThrdB*, and shared variable represents the interaction. If the shared variable is part of local state of each of the interacting agents then each of the threads is modeled as an active agent. However, we can consider the shared variable to belong to the local state of only one agent. In that case, the agent having the shared variable as part of its local state may be modeled as passive or active, while the other agent must be active (if the other agent is not active then there would not be any interaction). Note also that the interaction is about change in a state but not about data transfer as is the case with a communication protocol. The important difference between interaction and communication protocols is in the level of abstraction where the change happens: In the case of data transfer, the change is always in the value of data. However, in the case of interaction, the change can be in data, information, knowledge, or some other interpretation. Note that different protocols represent different kinds of possible interactions between agents. Also, different interaction protocols may be interpreted at different meta levels. Now, let us examine the case where the two threads communicate some data from *ThrdA* to *ThrdB* in a send-receive fashion such that *ThrdA* is sending while *ThrdB* is receiving data. The communication protocol may be like this: Initially, value of the shared variable is zero; *ThrdA* sets new value in the shared variable; *ThrdB* probes value in the shared variable permanently, and when the value is not zero *ThrdB* copies the value into some other place in its local state; After reading the value, *ThrdB* set value of the shared variable to zero; After setting new value, *ThrdA* has started to probe the value; After registering zero value in the shared variable, *ThrdA* knows that it is safe to set next value. In this case, at the interaction level both agents are active because each of them changes value of the shared variable. However, at the communication level, we say that the sender (*ThrdA*) is active while the receiver (*ThrdB*) is passive.

Definition 2: *Body of Knowledge (BK)* of an agent ag_i is defined as a part of its local state that is not observable $BK_i \equiv s(ag_i) \setminus s^O(ag_i)$.

Definition 3: *Link* is knowledge that an agent has about identity of some other agent. The link is knowledge that is sufficient for the agent to initiate an interaction protocol with the linked agent.

Definition 4: Let us given an interaction protocol, set of agents (called resources) and an individual agent (called agent) such that the agent can interact with the resources by the given protocol. *Addressing* (or *Code*) of the set of resources is a common service, such that there is guaranty that if an agent encounters the interaction protocol with different end addresses then it will interact with different agents, i.e. it may eventually experience different interaction histories.

In order to give an example for the previous definitions, let us consider an agent ag_{new} that has just been introduced into the universe of discourse. Since ag_{new} doesn't have any interaction history, it has empty body of knowledge, $BK_{new}=\varnothing$. Because $BK_{new}=\varnothing$, agent ag_{new} doesn't know about any links and is not able to activate any interaction. It has to wait for some other active agent to initiate interaction with the new agent. After finishing an interaction, it is expected that ag_{new} may have remembered[3] something from the previous interaction such that it may now have $BK_{new}\neq\varnothing$. If ag_{new} have learned address of some other agent during its last interaction, then ag_{new} may be able to initiate interaction with the agent on that address.

[3] An agent may or may not remember interaction histories depending on its internal memory resources. However, taking this into consideration is definitely out of scope of the paper.

Tag Cloud System

In this section, based on the previously defined concepts of resource, link, interaction, and knowledge, we present our understanding of the collaborative tagging as a set of concurrent acts of introducing new links into the system.

Definition 5: *Tag Cloud (TC)* is a tuple $TC = (R,L)$ where $R \subset U$ is a non empty *set of resources* contained in an *universal space of resources* U, also referred to as the universe of discourse, $L = \{(r,RID(p)) \mid r \in R, p \in U\}$ is a set of links, $RID(p): R \rightarrow A$ is a *resource identity function* that is mapping from the set of resources to the set of addresses A.

Note that the previous definition introduces a purely abstract category of resource as a member of the set of resources R and by means of the resource identity function *RID*. We say nothing about what the resource is, what is it's nature, structure, behavior or else. At this point, there is no semantics assigned to the resource. Instead, the resource can be anything that participates as a source of a link in the *TC*. The set of addresses A may be subset of a language or a subset of an enumerated set. The fact that set of addresses A is a subset of a language (or enumerated set) should be interpreted such that not every correct language construction is an address in the *TC*.

The *TC* represents a distributed knowledge system in a sense that we may consider something as a resource only after we learned about it as a resource. Similarly, we may consider a correct language construction as an address only after we learn about it as an address of a resource in the *TC*. On the other hand, the only way we can learn about new resources is to interact (inspect) with resources that are participating in our current knowledge. Further, In other words, we cannot speculate about anything that is not linked to at least one resource from the *TC*. In that way, we may say that *TC* represents the *Resource Universe*.

The natural interpretation of the TC is set of agents, as is introduced in Definition 1. We indicate this fact by the requirement that set of resources in the tag cloud is subset of the universe of discourse. In this way, we apply developed semantics of the multi-agent system to the tag cloud.

Definition 6: *Tag Cloud System (TCS)* is a tuple $TCS = (R,L,\Sigma)$, where $TC = (R,L)$ is a Tag Cloud, and Σ is a set of collections of resources from U such that each *collection* $C \in \Sigma$ is defined by the associated membership function *mC*.

The Tag Cloud System is a fine extension of the Tag Cloud structure that allows us to introduce collections into the Tag Cloud. The collection is defined by means of its membership function, with no constraints made on the function. The idea here is to have flexibility to being able to introduce different collections with membership functions of different nature, including fuzzy sets [17,18]. For example, in order to define set of tags R as a collection in the universe of discourse U, we use the membership function from the classical set theory: $mR: U \rightarrow \{0,1\}$, where $\forall u \in U$, $mR(u)=1$ if $u \in R$, and $mR(u)=0$ otherwise. However, we are not constrained to use such classical (or crisp) sets only. We can also use fuzzy set, which is a more general concept then the classical set: The membership of an element to a fuzzy set is not described by a Boolean function (as it is the case for a classical set), but by real values between 0 and 1, in general [18] (note that it can also be any other function, including discrete functions).

As we mentioned before, in the interpretation of the TCS, one cannot reason over anything else other then agents knowledge BK_i. Consequently, the membership function for an agent ag_i must be defined over BK_i only, for any collection under consideration. Typically, the BK_i includes addresses of other agents that are pointed to by links starting at agent ag_i. However, we do not put any restriction on the type of knowledge that may constitute BK_i. Thus, in addition to the "network topology" knowledge, an agent may have a free form text (for example, comment of the user who has created the agent while tagging some information on the Web), pictures, and any other type of structured or un-structured data. Later on, this knowledge is used for search or information extraction or any other purpose. The important advantage is that we integrate network topology information and free form information such that it can be queried in a unified manner.

Now, we introduce projection between two collections as a binary operation $\downarrow$ in the set of collections of a TCS.

Definition 7: Collection $C=C1\downarrow C2$ is *projection* of the collection $C1$ on the collection $C2$, defined as $C=(C1\downarrow C2)\equiv\left(\bigcup_{o\in C1}Chdrn(o)\right)\cap\left(\bigcup_{o\in C2}Chdrn(o)\right)$, where $\cup$ is (fuzzy) union and $\cap$ is (fuzzy) intersection operations

The projection of two collections, is defined with an aim to capture semantics of the TCS in the following way: First, find all tags of all resources from *C1*. Then find all tags of all resources from *C2*. Finally, find the intersection of the two sets of tags. The resulting set of tags should interpret "similarity" between collections *C1* and *C2*.

Resource Class in the TCS

Traditionally, we define class as *a collection of objects featuring some common (set of) feature(s)*. Following the previously introduced definitions, we may introduce class into the TCS in a similar way:

Definition 8: ***Class*** *C* in $TCS=(R,L)$ is tuple (O,T) where $O\subseteq R$ is a collection of resources, called ***objects***, and $T\subseteq R$ is a collection of (meta)resources, called ***tags*** (or features), such that every object $o\in O$ has identical projection of the collection of it's children into the given collection T: $\forall o1,o2\in O\ (Chdrn(o1)\downarrow T)=(Chdrn(o2)\downarrow T)$.

More informally, the set O is interpreted as a set of objects belonging to the class C. Set T is a subset of the set of all resources (resource universe) such that it's elements are identifiable as assigned semantics of being features. In other words, set T is subset of tags. However, we have to have a method to identify single resource as a tag. We implement this identification such that we have defined the page *Tag* with assigned semantics that *every resource R that has incoming link from the page Tag is perceived as being a tag*. The page *Tag* has a link to itself meaning that it is also tag.

Conclusion, Application Aspects and Future Work

To the best of our knowledge, the work presented in this paper introduces a theoretical model for semantics of the collaborative tagging systems, for the first time. We set the foundation for further exciting developments, particularly towards overcoming the gap between tagging as a Web 2.0 and tagging as a Semantic Web. The underlying model of knowledge-based multi-agent system has proven to be very helpful for us in solving practical application problems that show up during development of our tagging application prototype. In the prototype[4], we adopt and implement *Resource* and *Link* concepts. In that way, we got the unified, technology transparent, Semantic-Wiki-Tagging system. For example, according to Definition 3, each tagging contains link to the date when this tagging has been performed. However, we do not need to create an actual Wiki page for every such a date: agents in the system ('Wiki pages containing tagging data') have knowledge about identity of the date in a form of a Link. From the other side, the only interaction that may be initiated with the date is 'create page' because the date page is not able to engage into 'view page' interaction. Our future research will be to address the theoretical formulation of similar issues of the working prototype in more details.

The short indication given in the last section is particularly promising for future research. Definition of a tagging framework, similar to Object Oriented Programming, would definitely empower a whole new application space. One of the future challenges would be an object behavior within the TCS semantics. It is an open question whether a collection of resources is a resource itself (has an address or URI) or not. The similar problem exists with blank nodes in RDF [14,21]. Hence, we expect solutions similar to the one presented in [21] to be effective in the case of TCS too.

In this paper, we introduced the theoretical foundation for addressing different aspects of the semantic scripting by considering the collaborative tagging as a low-level scripting language on the global computational services fabric called "Web as a platform"[5]. We were focused on the system aspect of the semantic scripting. Depending on the level of abstraction[6], the target application may be traditional (collaborative bookmarks, annotations, etc.) or semantic (semantic Wiki, semantic web portal, semantic e-mail, etc.) or something completely innovative and new (such as tag clustering, tag hierarchies, tag cloud management, weighting and sequencing of tags, etc.).

---

[4] Code base of the prototype initially started as a modification of JSPWiki open source Wiki engine [19]. However, in time they developed into two almost independent applications.

[5] Analogous to shell scripting on an OS platform

[6] Level at the semantic web stack [20]

We are developing several Web applications based on collaborative tagging paradigm described in this paper. The current tagging application prototype can be accessed for testing at http://infosys-work.elfak.ni.ac.yu/InfosysWiki-v2-1/Wiki.jsp?page=TagCloud

References

1 Dan Connolly, Untangle URIs, URLs, and URNs: "Naming and the problem of persistence," *IBM developerWorks,* http://www-128.ibm.com/developerworks/xml/library/x-urlni.html
2 Tim Berners-Lee, James Hendler and Ora Lassila, "Semantic Web", *Scientific American*, May 2001.
3 Tim O'Reilly, "What Is Web 2.0, Design Patterns and Business Models for the Next Generation of Software," Sept. 30. 2005, Oreillynet.com, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
4 McFedries, P., "Folk Wisdom", IEEE Spectrum, Feb. 2006., pp.80-80.
5 Heather Richter, Chris Miller, Gregory D. Abowd, and Harry Funk. "Tagging Knowledge Acquisition To Facilitate Knowledge Traceability," in *Proceedings of the Conference on Software Engineering and Knowledge Engineering (SEKE)*, July 2003, pp432-439.
6 Golber, S., Huberman, B.A., "The Structure of Collaborative Tagging System", Information Dynamics Lab: HP Labs, Palo Alto, USA, available at: http://arxiv.org/abs/cs.DL/0508082
7 Lambiotte, R., Ausloos, M., "Collaborative tagging as a tripartite network", arXiv:cs.DS/0512090 v2 29 Dec 2005.
8 ACM Computing Classification System, http://www.acm.org/class/1998/
9 Champin, Pierre-Antoine; Jérôme Euzenat; Alain Mille: "*Why URLs are good URIs, and why they are not*", May 2001, http://www710.univ-lyon1.fr/~champin/urls/
10 Adam Mathes, "Source vs. Resource Ontology" http://www.adammathes.com/academic/krfo/sourceresource.html
11 Clark, Kendall Grant: "Identity Crisis," *XML.com*, September 11, 2002, http://www.xml.com/pub/a/2002/09/11/deviant.html
12 S. Pepper and S. Schwab. "Curing the Web's Identity Crisis," *Technical report, Ontopia* http://www.ontopia.net, 2003.
13 Berners-Lee, T.; R. Fielding; L. Masinter: "Uniform Resource Identifiers (URI): Generic Syntax and Semantics", *RFC 2396*, August 1998, http://www.ietf.org/rfc/rfc2396.txt
14 Hayes, P., "RDF Semantics – W3C Recommendation", http://www.w3.org/TR/rdf-mt/, 2004.
15 Wordnet, http://www.cogsci.princeton.edu/~wn/
16 Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi, "Knowledge-based programs," *Distributed Computing,* 10:4, 1997, pp. 199-225., also at http://www.cs.cornell.edu/Info/People/halpern/abstract.html
17 L.A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338--353, 1965.
18 Florea, M.C., Jousselme, A-L., Grenier, D., Bossé, É., "Combining belief functions and fuzzy membership functions," Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2003. Edited by Dasarathy, Belur V. *Proceedings of the SPIE*, Volume 5099, pp. 113-122, 2003.
19 http://www.jspwiki.org
20 Tim Berners-Lee. Web for real people, 2005. Available at http://www.w3.org/2005/Talks/0511-keynote-tbl/.
21 Carroll, Jeremy J.; Bizer, Christian; Hayes, Pat; Stickler, Patrick, "Named Graphs", *Journal of Web Semantics,* Vol. 3, No. 4, 2005, available at http://www.websemanticsjournal.org/ps/pub/2005-23