User Interface Design Considerations for Linked Data Authoring Environments

Stephen Davies, Jesse Hatfield, Chris Donaher, Jessica Zeitz
University of Mary Washington
1301 College Ave
Fredericksburg, VA 22401
1-540-654-1317
{sdavies, jhatfiel, cdonaher, jzeitz}@umw.edu

ABSTRACT

If non-technical end users are to contribute to the Web of Data as they have to the Web of Documents, they must employ tools that enable them to do so. This challenge is not easy to meet, as formal knowledge representation is a daunting task for the uninitiated. Indeed, we have empirically observed that expressing anything but the most straightforward of facts in RDF-compatible format is extremely difficult for newcomers to do reliably.

This paper reports on a controlled experiment in which novices attempted to use a prototype Linked Data interface to both find and encode bits of everyday knowledge. The application presents a user-friendly veneer to the Semantic Web, manifesting the essential graph-based nature of the data model while shielding the user from the complexity of syntax. This allows us to study user behavior in attacking the deep, cognitive problem: breaking down knowledge into the triple-based structure required by RDF Linked Data. Our study sheds light on some of the key aspects of knowledge formulation that novices struggle with, and suggests several specific design approaches for Linked Data authoring environments that our experiment makes clear beneficially address crucial issues.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Interaction styles; H.5.4 [**Hypertext/ Hypermedia**]: User issues.

General Terms

Semantic Web, Linked Data, User Interface Design, Experimentation, Human Factors.

1. INTRODUCTION

A successful, global-scale Semantic Web presupposes large amounts of instance data available for machines to process. As Tom Mitchell summarized during his ISWC 2009 keynote address[14] there are essentially three ways to produce this: (1) humans entering structured information, (2) database owners publishing their data in RDF format, and (3) employing automated natural langauge processing techniques to "read" unstructured Web data.

One might suppose that the only major impediment to (1) is convincing the masses that they have an incentive to do this. But in addition to the issue of motivation, serious questions arise about novices' *ability* to generate Linked Data in the format required by the Semantic Web. Formal knowledge representation is difficult and error-prone for most non-technical people. It is a very different activity from writing in natural language, which is the way that most laypeople have contributed to the Web to date. Authoring Linked Data demands an unswervingly consistent naming scheme, an unprecedented level of exactitude, fluency with a new suite of concepts, and an adherence to a set of rigid and (to the layperson) seemingly arbitrary rules that run counter to the way most people think, let alone converse. Though some psychologists (e.g., [1,10,19]) have thought semantic networks to be reflective of the way human memories are encoded, one only has to watch a novice struggle with expressing even basic concepts in a graph-based knowledge structure to know that this activity is extremely challenging.

We believe that for non-specialists to be successful in contributing to the Web of Data, they must use tools designed to compensate for their weaknesses. The design of such tools should be informed by empirical studies that illuminate how target users actually go about generating Linked Data, so that strengths can be maximized, weaknesses complemented, and unfruitful trends redirected.

The immediate goal of the work presented in this paper is not so much to design the ultimate Linked Data authoring environment as to empirically verify which aspects of such environments might be beneficial or harmful. By studying user behavior under simulated conditions, and observing which specific aspects of the Linked Data authoring process prove to be obstacles, we illuminate the nature of the problem and offer experimentally driven guidance on how to make end users successful.

The remainder of this paper is organized as follows. First, we describe related work in user studies of knowledge formulation processes and tools. Then, we introduce OKM[1], the prototype Linked Data authoring tool used in our experiments, highlighting key features whose viability we focused on in our study. We then describe the nature of our usability experiment, and present and interpret a quantitative analysis of the results. Finally, we summarize our findings and make generalizations and recommendations for future interfaces to Linked Data applications.

LDOW2010, April 27, 2010, Raleigh, USA.

[1] OKM is a recursive acryonym which stands for "OKM Knowledge Management," and is pronounced as "Occam." The prototype application is open-source and publicly accessible at http://sourceforge.net/projects/okm.

2. RELATED WORK

A wide array of tools have appeared in the last several years to help users in the RDF generation process. These include everything from semantic wikis (e.g., Platypus[7], Semantic Mediawiki[11], IkeWiki[17]) to semantic annotation tools (e.g., Loomp[12], OntoAnnotate[20]) to RDF editors (e.g., OntoWiki[2], Tabulator[4], IsaViz[16]) to full-blown ontology management environments (e.g., Protege[15], Swoop[9]). With few exceptions, however, published reports on these tools have not included usability studies to evaluate their effectiveness, or to identify the cognitive barriers users may face when using them. The result is a body of literature that contains many innovative and potentially useful user interface ideas, but with no core set of principles whose effectiveness has been proven and which can guide further work.

We mention here two notable efforts which did include illuminating usability studies. One was conducted by Staab *et al.*[20], who performed an in-depth analysis of the behavior of nine experimental subjects who used the OntoAnnotate semantic annotation tool. Their primary measure was inter-annotator agreement; that is, the degree to which different participants independently annotated a page in the same way. Their conclusion, roughly speaking, was that novices to the Semantic Web, operating in a domain where they are not experts, will not in general produce high-quality structured knowledge, or at least not knowledge that agrees with one another. If nothing else, this confirms the difficulty of the problem laypeople face.

Noy, *et al.*[15], on the other hand, performed an experiment in which military domain experts used a version of Protege-2000 with domain-specific extensions in order to perform specific knowledge acquisition tasks. The structure of the knowledge base given to participants was very detailed, and comprised a precisely specified class hierarchy containing concepts (e.g., types of combat units) that participants used on a daily basis. Unlike Staab *et al.*'s, Noy *et al.*'s conclusion was optimistic: these domain experts, with 1-2 hours of training but no computer science background, were in fact able to effectively use a large knowledge base that concerned a domain with which they were intimately familiar. The contrast between these two studies' outcomes testifies to the impact that domain expertise and domain-specific tools can have. The subjects in Staab *et al.*'s study, who used a general tool on general subject matter, had much greater difficulty. Clearly the more challenging user interface problem is to equip novices with a tool that is not custom-tailored to any particular subject matter, but which facilitates the proper construction of valid Linked Data on any topic, even one in which users do not begin with expert-level conceptions.

The setting we explore is more reminiscent of Staab *et al.*'s study, since we are focusing on laypeople (not domain experts) who are tasked with formulating generalized, open-ended knowledge. Our work differs from each of these efforts in that we are examining the effect and usage of specific user interface features, with the goal of discovering how a general Linked Data editor would best be designed. In particular, we analyze user behavior in choosing resources versus literals to represent information, the efficacy of employing types and templates in the interface to steer users towards data consistency, and alternative ways to express n-ary relations. None of these UI aspects has, to our knowledge, been empirically studied in a focused, experimental setting.

3. OKM FEATURES

3.1 Basic Design

OKM's primary purpose is to serve as a testing bed for analyzing how laypeople interact with Linked Data tools, and its basic design is common to many state-of-the-art RDF and ontology editors. This commonality is key in relating OKM to tools currently in use by the Semantic Web community; with it, we hope to generalize the results we obtain from empirical testing to Linked Data authoring as a whole.

For instance, like OntoWiki[2], Tabulator[4], Kiwi[17], Semantic Wikipedia[22], and many other tools, OKM's pages are "resource-centric" in that each page represents a single resource, displaying all the properties relating to that resource. Hyperlinks to related resources can be used to traverse the site. As with Freebase[5], users primarily interact with the system in terms of human-readable names (HRNs) rather than full URIs. At resource creation time, OKM auto-generates a globally-unique URI for that resource (scoped to the domain name of the OKM server), but users continue to work with HRNs in order to diminish screen clutter and enable more focus on semantics than syntax.

Users can add datatype or object properties to a resource directly from its page. In the interface, OKM refers to datatype properties (whose values are literals) as "attributes" and object properties (whose values are resources) as "statements."[2] (We will use this terminology throughout the remainder of this paper.) The use of two terms (instead of calling everything a "triple") is intended to help the user better appreciate the distinction between them, since they are created, presented, and navigated differently. If the user chooses to add an "attribute," the property value will be interpreted as a primitive data type. If the user chooses to add a "statement," the property value will be interpreted as the HRN of another resource. For statements, the user can specify an existing resource in the system as the object – at which point the new resource is effectively "stitched in" to the rest of the graph – or else refer to a resource which does not yet exist, which will implicitly create that resource.

Users can also search the system for resources by typing in a search box that autocompletes based on HRNs, or any portion thereof (e.g., typing "lin" will match a resource whose HRN is *Abraham Lincoln*.) This functionality is of course common to innumerable tools today, from Freebase[5] to IsaViz[16] to non-semantic-web tools like Wikipedia and the Google search interface. Also, an explicit "create" box allows resources to be created from scratch, and not (initially) connected to anything.

Again, since this design is similar in spirit to that of many tools in existence today, we believe that empirical findings based on OKM's interface will be of broad interest to the community of Linked Data researchers studying user interfaces.

[2] We chose these words based on survey feedback from a previous experiment[8] in which users were asked for the most intuitive terms for the two concepts.

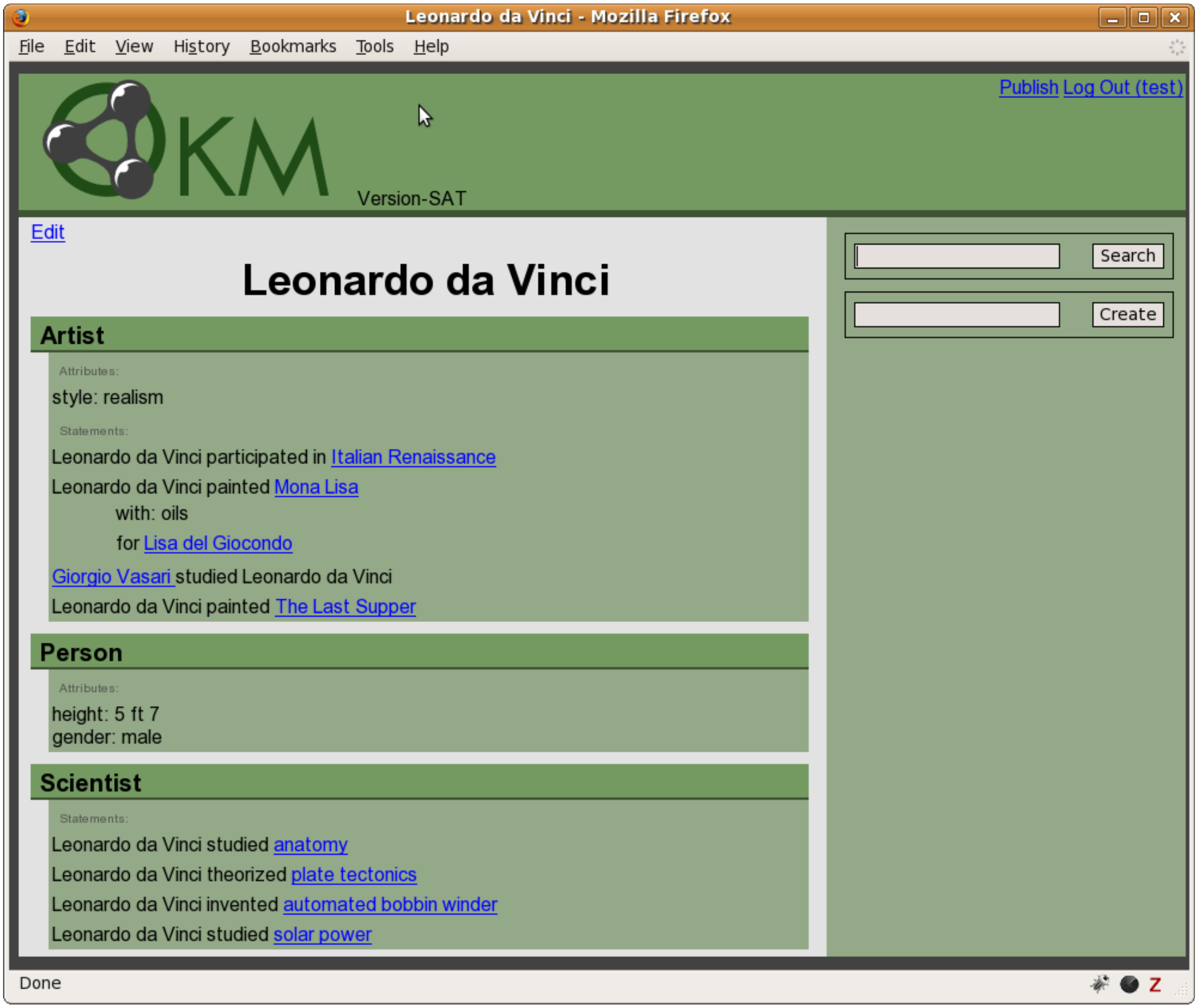

Figure 1. The basic OKM interface, in "view" mode. (SAT version.)

3.2 Linked Data Publishing

OKM stores all information that the user creates in a local Jena RDF triple store[13]. Appearing in the upper corner of every OKM page is a "Publish" link which, if pressed, will generate Linked Data for the currently-displayed resource in RDF/XML format. This Linked Data is stored in a file in a configurable location on the web server that is hosting the OKM installation. It can then be accessed over the Web by dereferencing the URI that OKM auto-generated for the resource, according to Linked Data principles.[3] Note that the RDF/XML file will contain a serialization of (1) all triples for which the currently-displayed resource is a subject, and (2) rdfs:seeAlso links for the URIs of resources that appear as the subject of a triple for which the currently-displayed resource is an object.

If the user presses the "Publish" link while viewing the OKM home page, Linked Data for *all* resources in the local system will be generated. The entire knowledge base will thus be globally exposed to Linked Data consumers.

In this way, Semantic Web amateurs can be empowered to contribute to the Linked Data movement by utilizing a tool with a low barrier to entry and which shields them from the syntactic complexities of RDF. Note that the current version of OKM does not support "round-trip" knowledge creation whereby existing Linked Data (and ontologies) can be imported into the tool. This feature was postponed since it did not bear upon our immediate experimental concern; in future studies, however, we plan to implement this and study user behavior in interacting with a larger, pre-existing knowledge space (in which there is greater urgency to find and re-us existing resources.)

3.3 Experimental Features

Supplementing this normative user interface are three atypical features, which formed the focus for most of the investigative effort described in this paper. We hypothesized that each of these changes to the pseudo-standard user interaction paradigm would prove beneficial to novices attempting to interact with Semantic Web data, and for different reasons.

3.3.1 Roles and Templates

Rather than presenting all properties of a given resource in one long display, OKM encourages – and in fact, mandates – organization of these properties according to the resource's "roles." A role is essentially an rdfs:Class to which the resource belongs, and which acts as the rdfs:domain (or rdfs:range) of the properties relevant to that class. Consider the screenshot in Figure 1. Here, the "Leonardo da Vinci" resource (which of course has a unique URI but which is presented to the user in terms of its HRN, as described above) has three roles: Artist, Person, and Scientist. Each role is manifested as its own box, with the relevant statements as contents. A given triple about Leonardo Da Vinci will appear in the role box which represents the domain for that triple (or, if Leonardo Da Vinci is the object rather than the subject of the triple, in the role box which represents the range. The "Giorgio Vasari" triple is an example of this latter case.)

In order to add an RDF triple to the system, the user *must* choose one of the resource's roles (or add a new role) which will serve as the domain of the triple, and then add the triple in the corresponding role box. The user begins this process by clicking on the "Edit" link at the top of the page, thereby putting the page in "edit mode." (See Figure 2 for an example.) The role boxes then acquire buttons labeled "+Attribute" and "+Statement," which can be used to add attributes or statements to that role box. The user can then type the name of a predicate and a value. An autocomplete function assists the user with both inputs, offering to match predicates already in the system, and (in the case of statements) HRNs of resources already in the system. It is perfectly permissible, however, for the user to type the name of a new predicate and/or the name of a new resource, in which case the new item is implicitly created. The new predicate is automatically given a domain based on the role box it was added to, and a range based on the role of the object value it was given. (For object resources with multiple roles, the "Set Role" button can be used to select which of the resource's roles should be the range of the predicate.) From that point forward, the system incorporates the new predicate into its ever-evolving schema.

One important aspect of roles is that when in edit mode, a *template* appears within each role box that displays the predicates already known to have that role as a domain. Using these templates is similar to inserting data in Freebase's type-based editing model [6]. In Figure 2, note the predicates "dimensions," "period," and "influenced" which appear in grey. These predicates – which are absent when the resource is being seen in "view mode" – appear in the box because at least one other resource with the "Painting" role has a triple involving each of these predicates. Pressing the "Add Value" button next to a grey item will prompt the user for a value for that item. In this way, the template suggests to the user possible predicates that are consistent with the schema that exists thus far.

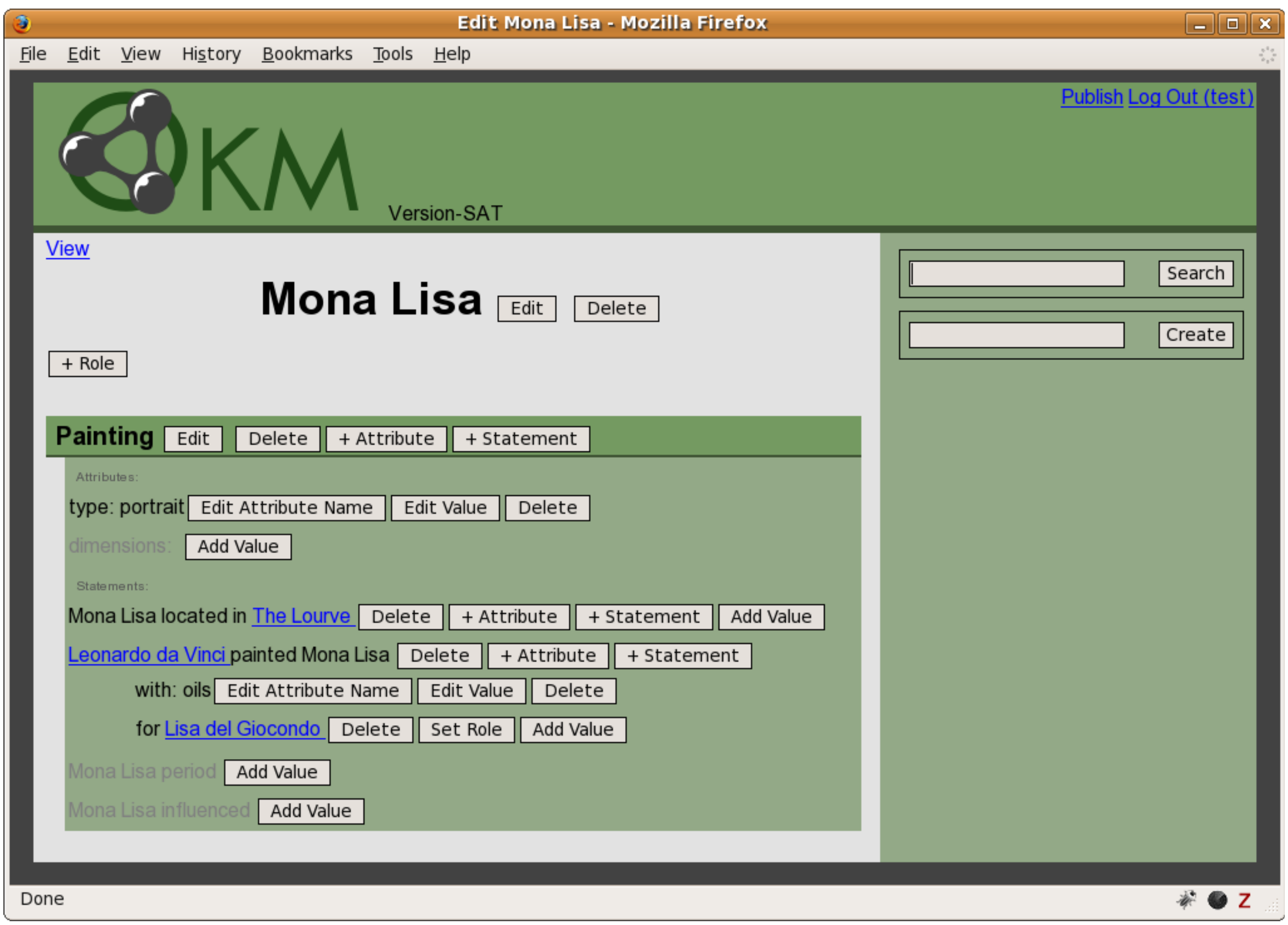

Figure 2. The basic OKM interface, in "edit" mode. (SAT version.)

In any fairly complex knowledge base, we can predict that most users will be unable to keep track of all the predicates in use and will inevitably use different predicates to represent the same semantic concept. OKM's templates are designed to help guide users into editing resources in such a way that they stay within the current schema, while not constraining users from adding to that schema.

In summary, then, roles are intended to provide three benefits:

1. They lend organization to the display when a resource has many triples, in order to make information easier to find and enter.
2. They ensure richer data (with domain, range, and type information) than novice users would ordinarily produce. It seems likely that when authoring only simple triples, most novice users would not bother to assign types to their resources, nor domains and ranges to their predicates. (At the least, it is unlikely that they would consistently do this.) With OKM, however, the act of assigning types, domains, and ranges is built in to the very process of creating triples, making it convenient to do and impossible to avoid.
3. They provide a template of relevant predicates that is easy for users to fill out. This provides not only instantaneous ease of use, but promotes long-term data consistency.

We present benefits 1 and 2 without proof. Later in this paper, we provide an in-depth empirical analysis to judge the efficacy of benefit 3.

3.3.2 The Elimination of Attributes (Literals)

The flexibility that RDF offers in supporting both resources and literals as object values is a mixed blessing. On the one hand, it presents an expressive modeling device. The objects of the triples "John marriedTo Sally" and "John weightInPounds 175.4" seem inherently different: "Sally" is presumably a bona fide resource in her own right, with other triples expressing information about her, whereas "175.4" intuitively seems like a primitive piece of raw data, undeserving of resource status. Allowing authors to designate an object as one or the other affords the opportunity to express this subtle aspect of the object.

On the other hand, the existence of the distinction means that authors are *forced* to choose between the two alternatives, and the choice is not always easy to make. Consider triples like "BeverlysToyota color red," "Charlie bornIn 1982," and "Candice schoolYear sophomore." The object values "red," "1982," and "sophomore" might be considered literal pieces of data, as with the above weight example, or as first-class resources. Anyone who has composed RDF for any length of time knows that this choice presents itself at every turn, and that in some cases it feels almost arbitrary.

Our work presents two contributions toward better understanding this phenomenon and how to best handle it. First, by creating a system that lowers the barrier of entry for the creation of RDF, as well as a system for creating both statements and attributes, we can observe how uninitiated users tend to differentiate between the two in practice. Later in this paper we present findings that reveal user tendencies in choosing between statements and attributes for specific types of information, and an analysis of the degree of consistency laypeople exhibit in this choice.

Second, we explore the effects of an RDF editor in which attributes are simply *eliminated*. It is possible, of course, to completely do away with the concept of literals if one is prepared to accept elements like "175.4" as resources. This is one way of dispensing with both the angst users face in making the decision, and also the inconsistency that can result when users make different choices: simply take away the choice altogether. This may seem like a heavy-handed solution, but it is not without theoretical merit. Consider that more than one prominent cognitive psychologist (e.g., [1, pp.125-7; 10, pp.34-92; 23]) has formulated a knowledge representation theory based on something akin to semantic networks, yet found no need to differentiate between resouces and literals. One kind of node is all that comprises these knowledge structures, which suggests that a "resources only" network is in fact sufficient to encode human knowledge. And it places the burden of proof rather on those who argue for the existence of two distinct kinds of entities.

As described below, we deployed to experimental subjects not only the version of OKM depicted in Figures 1 and 2, but also versions in which attributes were completely eliminated. The "+Attribute" button was removed from all displays, which effectively forced users to model everything as statements. We then compared accuracy, consistency, and user satisfaction between the different versions.

3.3.3 Predicate Modifiers

Lastly, OKM allows users to construct n-ary relations without explicitly using reification. This feature was inspired by a recent project in which we conducted a pencil-and-paper based experiment[8]. In this study, young adults with no previous Semantic Web experience were asked to construct knowledge representations (both visually and textually) corresponding to English sentences. Some of these sentences contained facts which were inherently n-ary: "Muhammad Ali fought Joe Frazier in Detroit," for example. (This statement relates three entities and hence cannot be expressed as simple subject-predicate-object triples without reifying the verb.) Our participant pool was divided so that half of them were shown solutions to such sentences using traditional reification techniques: first, create a resource representing the verb ("AliFrazierDetroitFight," perhaps) and then attach the other resources to it with predicates like "participant" and "location." The other half of the participant pool was instead shown solutions involving predicate modifiers: they were permitted to break outside the strict triple scheme and augment a triple with further information indented beneath it. (This is the scheme supported by the Yago knowledge model[21].) To illustrate, users could express the above sentence textually as:

```
MuhummadAli  fought  JoeFrazier
       in  Detroit
```

This is really nothing more than a shorthand notation for treating the first triple as the subject of a second triple, but it proved to have an enormous impact on user success. (As an example of the size of the effect, for one of the items 62% of participants were able to correctly express the sentence using predicate modifiers, compared with 14% using traditional reification.) The overall conclusion is that end users can be far more successful in constructing n-ary relations when enabled to employ predicate modifiers than when they are forced to express them as reified triples.

Guided by these findings, we implemented a predicate modifier scheme in OKM. An example is the "Leonardo da Vinci painted Mona Lisa" fact in Figure 1. Note that "with: oils" is an attribute, and "for Lisa del Giocondo" is a statement, and that both are *indented* underneath the "painted" triple. Users create such indented facts by pressing the "+Attribute" or "+Statement" buttons *next to a triple,* rather than at the top of the role box (refer to Figure 2.) When generating Linked Data, OKM converts these indented facts into traditionally reified triples, so that the knowledge is compatible with all current Semantic Web tools. From a user interface perspective, however, users never see the complexities of reification: they view and edit n-ary relations in terms of the much more intuitive predicate modifiers.

4. EXPERIMENT

4.1 Hypotheses

We formulated the following hypotheses to evaluate experimentally.

Regarding the **"roles and templates"** feature:

- H1A – The addition of a "roles and templates" feature will significantly increase laypeople's ability to *correctly* formulate Linked Data: i.e., the RDF they generate will make more sense semantically.
- H1B – The addition of this feature will increase the likelihood that laypeople will *consistently* formulate Linked Data: i.e., they will more often reuse appropriate predicates that already exist.
- H1C – Users will in general employ the role feature properly by *selecting appropriate roles* (and thus incidentally contribute meaningful domain, range, and type information.)
- H1D – Users will in general *select roles consistently* with one another (i.e., if two users separately encode the same bit of knowledge, they are very likely to select the same role under which to create the triple.)

Regarding the **"elimination of attributes"** feature:

- H2A – If given the choice of creating an attribute or a statement for a given bit of knowledge, there will be *no predictable consensus* among of a group of laypeople. They will very often make inconsistent choices with one another, leading to gross inconsistencies in a collaborative knowledge base.
- H2B – Laypeople who employ a Linked Data interface that *eliminates attributes* altogether will suffer *no disadvantages*: the data they generate will be as correct as those who have both statements and attributes available.

Regarding the **"predicate modifiers"** feature:

- H3A – Given examples of both predicate reification (traditional) and predicate modifiers (as described above), and the choice to use either technique to express n-ary relations, users will choose the latter significantly more often, and have more success in doing so.
- H3B – The presence of the predicate modifier feature will have no significant negative impact on laypersons' knowledge generation: i.e., it will rarely if ever be misapplied to produce errant knowledge.

4.2 Participants

Our participant group consisted of 71 college students ranging from 18 to 22 years of age and contained roughly an even split between genders. All students were enrolled at the University of Mary Washington during the Spring 2010 semester and were of many diverse majors.

4.3 Procedure and Materials

Participants took the one-hour experiment using the Firefox Internet browser on either a Windows or UNIX workstation. A ten-minute demonstration and explanation of OKM was given, and then each participant received an experiment packet and was directed to a URL (unique for each participant) that housed an OKM deployment with a pre-fabricated knowledge base containing about 130 resources and 150 predicates. The packet included 10 questions to be answered using this knowledge base (Part 1) and 24 facts to be added to it (Part 2). The final part of the packet (Part 3) was a survey to help us better analyze how the participants reacted to the system.

Part 1 questions ranged from easy to difficult depending on how difficult it was to find the information in the system. Easy questions were ones where the participant had to locate a specific resource page in the system and the answer was directly on that page. For example, "How tall is Jason Thompson?" The "Jason Thompson" resource existed in the system and the answer could be found on that page. More difficult questions forced the participant to view multiple pages and traverse links within the pages to locate the answer. For example, "What ballpark does Todd Helton's baseball team play in?" The participant had to first find the "Todd Helton" resource page, and then find and click the link to the "Colorado Rockies" page in order to find the name of the sports facility in which the team played. Part 1 also acted as practice to help the participants become more comfortable and aware of the system and how it was organized.

Regarding hypothesis H3A, it is important to note that the last two Part 1 items involved n-ary relations, but that the pre-fabricated knowledge base had encoded one of them using predicate modifiers, and the other using predicate reification. The two items had nearly identical structure: "For what novel did Ernest Hemingway win the Pulitzer Prize?" and "For what film did Martin Scorcese win the Academy Award for Best Director?" Hence in answering this question, all participants witnessed properly encoded examples of both predicate reification and predicate modifiers. They were then presumably not biased in either direction when beginning Part 2, which required the encoding of five n-ary relations among its 24 items.

Part 2 had the participants add data to the knowledge base. This part had a range of difficulty levels just as Part 1 did. The facts were presented as sentences with each sentence having one to three small facts within it. For example, "Madison Square Garden is located in New York City" has one fact: the fact that Madison Square Garden is in New York City. "Mark David Chapman assassinated John Lennon on December 8, 1980 at the Dakota Apartment Complex," on the other hand, has three facts: the fact that Mark David Chapman assassinated John Lennon, and the date and the place of the assassination. (Note that this is an n-ary relation.) Resources referred to in part 2 did not always exist in the pre-fabricated knowledge base, requiring the participant to create a resource before adding the fact.

Our participants were split into four groups based on which version of the program they used. The four versions were:

ST (18 participants) – A "*statements only*" interface (i.e., no attributes) that provided role-based *template* information when editing resources (as described above.) This is the version of the interface which we hypothesized would be the most effective, since it incorporated all three of the experimental features described above.

S (19 participants) – A "*statements only*" interface with *no* templates. This version was identical to ST, except that when in edit mode, the greyed-out "suggested" predicates (such as "dimensions," "period," and "influenced" in Figure 2) would not appear.

SAT (18 participants) – A "*statements + attributes*" interface *with templates*. This version was identical to ST, except that the "+Attribute" buttons were included so that users could choose between creating attributes or statements. (This is the version of the interface depicted in Figures 1 and 2.)

SA (16 participants) – Finally, in order to test hypothesis H2A, a number of participants received a more "traditional" version of OKM that permitted both statements and attributes, but provided no templates.

We then evaluated our hypotheses by judging the contents of the Linked Data knowledge bases that users produced while carrying out the actions required in Part 2. We did this in the following way:

- H1A – compare groups S and ST for correctness.
- H1B – compare groups S and ST on the items for which an appropriate predicate already existed in the pre-fabricated knowledge base, to determine whether they used that predicate.
- H1C – judge all groups on how often the roles they chose to put a triple under was conceptually correct. This was admittedly somewhat subjective, but in practice there was very little debate among the graders (the four authors of this paper) as to whether a role was correct.
- H1D – for each item, evaluate the frequency with which participants chose the same role using Simpson's diversity index[18].
- H2A – for participants in groups SAT and SA, evaluate the degree of consensus participants exhibited in choosing attributes or statements to represent the information.
- H2B – compare groups SAT and ST for correctness.
- H3A – for the five Part 2 items requiring n-ary relations, count the number of times participants (in all groups) used predicate reification versus predicate modifers to correctly encode them.
- H3B – for the nineteen Part 2 items that did *not* require n-ary relations, count the number of times participants mistakenly used the predicate modifiers feature and generated nonsensical Linked Data as a result.

5. RESULTS

5.1 Results: Roles and Templates

5.1.1 The Effect of Templates

We evaluated our template-related hypotheses using nine specific items. For each of these items, the system contained a predicate, associated with an appropriate role, that users should have noticed and could have selected using the template. However, these items were in two groups. For six of them (items J, K, L, P, W, and X) the already existing predicate was in fact semantically appropriate for the item. For instance, for item J, "David Beckham scored 27 goals," the pre-fabricated knowledge base contained the predicate "goals" for the "Soccer Player" role. Therefore, it would have been appropriate and consistent for ST users to select "goals" from the template to record this item (as opposed to creating their own equivalent predicate such as "scored" or "numberOfGoals.")

For the other three items, however, the already existing predicate was *not* semantically appropriate for the item. We called these items "traps." For example, the pre-existing predicate that ST users saw for item O, "The Matrix's gross earnings were $90 million," was "*net* earnings." We included these three items because we wanted to measure the degree of danger templates may introduce in leading users to choose pre-existing predicates that are in fact not appropriate.

We evaluated templates in two ways. First (hypothesis H1A) we compared the total correct responses between the ST and S groups for the nine items, treating each users' response for each item as a seperate trial. Considering all nine, 70.37% of ST users' representations (114 out of 162) were correct, as opposed to 67.25% (115 out of 171) of S users' representations. This difference was not statistically significant ($p > 0.01$ by Fisher's exact test). This indicates that templates do not improve *correctness* of user-generated data, thereby refuting hypothesis H1A. (Interestingly, when considering only the six non-trap items, the ST group got 65.74% correct compared with S's 61.40%; for the trap items, ST got 79.63% and S 78.95%, neither of which was statistically significant. It appears, then, that templates have no impact on correctness, regardless of whether the predicates in question are in fact appropriate.)

We also studied the impact templates have on *consistency* of data (H1B); i.e., the likelihood that data authors would re-use an appropriate predicate already existing in the system as opposed to creating a synonymous one. The effects of templates on the predicates used for the nine items are summarized in Table 1.

Table 1. Effect of role templates on predicate usage.

	ST Predicate used:		S Predicate used:		P-value
Item	Existing	New	Existing	New	
J	7	11	0	19	0.0031
K	10	8	0	19	0.0001
L	15	3	4	15	0.0002
P	1	17	0	19	0.4865
W	5	13	1	18	0.0897
X	3	15	0	19	0.1050
(Traps:)					
C	18	0	19	0	1.0000
O	12	6	19	0	0.0080
V	17	1	19	0	0.4865

The results show that users in the ST group were significantly ($p < 0.01$ by Fisher's exact test) more likely to (correctly) use the existing predicate in three of the six non-trap cases, and to (incorrectly) use it in one of the three trap cases. This is a mixed result. Evidently, templates effectively promote consistency for some facts but not for others, and they mislead users into semantic incorrectness for some facts but not for others. Hypothesis H1B appears to be confirmed only in certain cases.

When we probe the specific items to discover which *kinds* of items templates assist, we discover that it greatly depends on word choice. Templates were not shown to be helpful for items P ("The song 'Stairway to Heaven' featured Jimmy Page on guitar", with predicate "plays" defined for role "Musician"), W ("John Entwistle played bass on the song 'Behind Blue Eyes'", with predicate "plays" defined for role "Musician"), and X ("Paul McCartney wrote the song 'Maybe I'm Amazed'", with predicate "composed" defined for role "Musician"). The wording of items P and X differs from the defined predicates, while the tense differs for item W. These results suggest that users are less likely to use templates when it would require restructuring the sentence or using different terminology.

In a real-world setting, of course, users are not translating sentences into Linked Data, but "mental knowledge" into Linked Data. We can only speculate as to the size of this effect for mental knowledge, but it seems reasonable to assume that if a user wants to encode a fact, and has a certain phrasing in mind, they will succumb to the same pitfall that our testers did.

Note that two requirements must be met in order for a user to take advantage of a template: they must (1) select the proper role (i.e., the role that is the domain for the predicate), and they must (2) observe and decide to use the relevant predicate for that template. In cases where the user failed to use templates, neither of these factors was entirely to blame. For items P and J, for example, 81.8% and 82.4% of the failures were due to choosing the wrong role; for items W and X, on ther other hand, 100.0% and 86.7% were due to not using the right predicate within the (correct) role. It appears possible to fail at either.

Item O ("The Matrix's gross earnings were $90 million") illustrates the potential negative impact of templates. Six out of eighteen ST users chose to use the existing predicate "net earnings." This result indicates a risk that users may select incorrect predicates when they are lexically similar but semantically different from the phrases they intend to use. However, this risk can be expected to diminish with substantial domain knowledge, which equips users to correctly differentiate between similar terms.

Overall, these findings suggest that templates assist with both consistency and correctness in predicate usage, despite not being effective in all cases. Templates are helpful when the user selects the appropriate role and when the existing predicate is consistent with the user's intended phrasing of the information.

5.1.2 The Viability of Roles

In order to evaluate the viability of roles, we examine both correctness (H1C) and consistency (H1D) of roles chosen for 20 items. We judged correctness by the relevance of the chosen roles to the information entered. For example, we considered appropriate roles for item L – "Deion Sanders has stolen 35 bases" – to include "Baseball Player" and "Athlete" but not "Football Player" or "Person." Inconsistency is measured using Simpson's Index of Diversity ($D = 1 - \Sigma(p_i^2)$, where p_i is the proportion of users who chose role *i*) for each item. An index of 0 indicates that all users chose the same role. Larger values indicate more distinct roles chosen and a more even distribution between roles. Both are shown in Table 2, divided into two groups: items for which the resource already had an appropriate role, and items for which the user would have to add a role to the resource in order to make a reasonable choice.

The results show that users were very likely (92.81%) to make a reasonable choice if the relevant objects already had an appropriate role, but less likely (73.32%) to add one themselves ($p < 0.05$ by a t-test). The diversity of roles was high for both kinds of facts, although the second group was more diverse ($p < 0.05$). This indicates that users are often unable to choose correct roles, and are not reliably consistent with one another. The fact that templates were successful at helping users enter semantically correct data despite these difficulties suggests that users better guided by ontologies might experience greater benefits from a system that incorporates type and schema information.

Table 2. Correctness and consistency of role choices.

Object Already Had Reasonable Role

Item	% Correct	# of Roles	D
D	98.51%	4	0.19
E	76.81%	3	0.36
H	95.59%	4	0.17
K	94.37%	4	0.13
M	100.00%	1	0.00
N	88.57%	16	0.65
P	98.53%	3	0.31
S	75.71%	16	0.67
T	88.73%	6	0.72
U	95.59%	4	0.53
V	98.55%	3	0.18
W	97.06%	3	0.06
X	98.57%	4	0.08
Average	**92.81%**	**5**	**0.31**

User Forced to Add Role

Item	% Correct	# of Roles	D
F	44.93%	4	0.53
G	69.23%	6	0.48
J	89.55%	6	0.58
K	92.31%	6	0.42
L	66.20%	9	0.53
O	75.36%	13	0.47
Q	77.14%	9	0.43
R	71.83%	4	0.53
Average	**73.32%**	**7**	**0.50**

5.2 Results: Elimination of Attributes

To determine whether the presence of attributes in the system influences users' ability to represent data, each item was rated for correctness. A response was considered correct if it accurately conveyed the information given in the text, was consistent with the graph-based data model, and was associated with a reasonable role. Table 3 compares the SAT and ST groups for overall correctness of data entry.

The results show no significant difference between the two groups (by Fisher's exact test, $\alpha = .01$). (It is also the case that no significant difference existed on any one item.) This finding is consistent with our hypothesis that the presence of attributes in the system would not influence the quality of data produced by users. Thus hypothesis H2B is confirmed.

We also hypothesized that, when forced to choose whether to model a fact as a statement or as an attribute, users would behave inconsistently with one another. Table 4 shows the data we collected to evaluate this hypothesis. The 24 sentences contained 30 atomic facts, which are divided into five categories based on whether their objects are: proper nouns (14 facts, such as "New York City"), common nouns (5 facts, such as "piano"), numeric values (9 facts, such as "186 pounds"), dates (3 facts, such as "April 4, 2008"), and years (1 fact, "2004").

Table 3. Impact of attributes on correctness.

Group	Correct	Incorrect	% Correct
SAT	336	96	77.78%
ST	353	79	81.71%
		p = .18	

Table 4. Consistency of the attributes vs statements choice among various types of data.

Category	Number of items	% attributes
Common nouns	5	27.10%
Date	3	56.50%
Numeric	7	74.30%
Proper nouns	14	16.20%
Year	1	20.70%

Our users demonstrated the highest consistency for proper nouns and the lowest for numeric values. However, it is clear that for none of the five types can consistency be counted on. No matter what type of fact is being represented, different novice users will encode it in different ways – some as attributes, some as statements – leading to basic inconsistencies in the resulting structure of the Linked Data.

Evaluating consistency in the abstract is difficult, but we note certain items that show a surprising lack of consensus. For instance:

Item F - "Kelly Witt is a freshman." (14 attrs, 18 stmnts)

Item S - "Michael Abram stabbed George Harrison on Dec. 30th, 1999" (the date: 19 attrs, 12 stmnts)

Item Q - "Deion Sanders hit 7 home runs" (24 attrs, 9 stmnts)

Users exhibit no strong consensus regarding how best to model these pieces of information, and many others, confirming hypothesis H2A.

For some items, a substantial number of users made very unintuitive choices. Item E, clearly a numeric value ("Ryan Medina's GPA is 2.79") was represented as a statement 11 out of 33 times (33.3%). Even more problematic is the tendency to represent proper nouns as attributes. For item A ("Madison Square Garden is located in New York City") 9 out of 30 users chose to represent New York City as an attribute (30.0%) And for item N ("Mark David Chapman assassinated John Lennon on December 8, 1980 at the Dakota Apartment Complex"), 17 out of 31 users represented Dakota Apartment Complex as an attribute (54.8%). We argue that representing proper nouns like "New York City," about which many things on the Semantic Web are likely to be said, as anything other than resources is a mistake, and that untrained users are likely to make that mistake often when given the choice. In the absence of a compelling reason to do so, we recommend that systems not force users to make the choice between resources and literals.

5.3 Results: Predicate Modifiers

To evaluate the effectiveness of predicate modifiers as a tool for expressing n-ary relations, we exposed users to both predicate modifiers and the traditional method of predicate reification. *None* of the 71 users employed predicate reification in representations of any of the five facts containing n-ary relations. Table 5 shows the overall correctness of the user's representations of those facts (using predicate modifiers).

Table 5. Use of predicate modifiers for n-ary relations.

Item	Correct responses (out of 71)
I	60.56%
N	66.20%
P	50.70%
S	56.34%
W	49.30%
Average	**56.62%**

Users were less likely to express these facts correctly than simpler items. However, in light of our previous study[8] that showed that people with minimal training are extremely unlikely to properly express n-ary relations using triples, the results are promising. Hypothesis H3A is soundly confirmed.

For the vast majority of test items, which contained only a single fact each, users did not attempt to (incorrectly) represent them using predicate modifiers. (No more than 2 out of 71 users tried this for any of those items.) However, for item K ("Peyton Manning passed for 206 yards, while Brett Favre threw for 315") this did prove to be a common problem. This item actually contains two separate binary relations, but 25 out of 71 total users (35.2%) incorrectly applied predicate modifiers to try and express it. This finding may suggest that novice users can have difficulty determining whether a complex thought represents a single n-ary relation, or a series of binary relations. If so, we argue that this only emphasizes the need to investigate more intuitive techniques for representing complex information. In any case, for simple sentences, hypothesis H3B is confirmed.

5.4 Results: survey

Finally, our experiment ended with a 12-question survey in which participants answered reaction questions on a 6-point Likert scale. These measured user satisfaction with the system, the ease with which they could locate information, etc. Only two of the items demonstrated any significance between groups (to an α of 0.05):

> "It was easy to use the system to add new information." The average responses on this item were: Group S=4.9, Group ST=5.5, Group SAT=4.1. Considering "templates" and "attributes" to be two independent variables, a univariate ANOVA test confirms that a "statements only" interface has a beneficial effect on user perception of how easy it is to add data. ($p < 0.05$).
>
> "I was confident that I added the information correctly." The average responses were: Group S=3.8 Group ST=4.6, Group SAT=3.9. The ANOVA test confirms that the template feature has a beneficial effect on user confidence in adding data. ($p < 0.05$)

(Note that since we had no survey information from an "SA" group – i.e., attributes, but no templates – we could not detect any possible interaction between variables.)

Although we had no *a priori* hypotheses regarding user preferences, this survey information seems significant. Users appear to have a preference for a “statements only” interface with templates. This type of interface modestly enchances the user experience and raises confidence.

6. CONCLUSIONS

Novice end users, who are potential contributors to the Web of Linked Data, have substantial difficulties formulating knowledge in the format the Semantic Web requires. User interface design, therefore, is paramount. Our empirical testing has shed light on certain aspects of how such interfaces are used in practice, and should be best designed. These include:

1. Requiring users to group information about a resource according its roles (types), and displaying a template of previously used predicates for each of those types, can help channel users towards the re-use of predicates that already exist, avoiding undesirable proliferation of synonyms. We also believe that roles and templates provide benefits in terms of more facile navigation and the implicit creation of domain, range, and type information. However, users are not consistent in their choice of roles, suggesting that some mechanism for encouraging consistency would be wise.
2. Users are very inconsistent in choosing to model an object as either a resource or a literal, and this appears to be true across a wide variety of types of objects. In many cases they simply make inappropriate choices. Moreover, users appear to be just as successful generating Linked Data when they use an interface that *only* supports resources. We therefore recommend that tools for authoring Linked Data not include literals in the interface.
3. A scheme for allowing users to express n-ary relations with modified predicates, rather than with traditional predicate reification, can enormously increase their success in modeling such information.

7. ACKNOWLEDGEMENTS

We would like to thank Trillane Burlar and Christopher (Shane) Voisard for their inimitable development work, without which this project would not have been possible.

8. REFERENCES

[1] Anderson, J. *Cognitive Psychology and its Implications (Seventh Edition)*, Worth Publishers, 2009.

[2] Auer, S., Dietzold, S., and Riechert, T. “OntoWiki-A Tool for Social, Semantic Collaboration,” *Lecture Notes in Computer Science*, vol. 4273, pp. 736-749, 2006.

[3] Berners-Lee, T. “Linked Data – Design Issues,” 2006. Available at: http://www.w3.org/DesignIssues/LinkedData.html.

[4] Berners-Lee, T., Hollenbach, J., Lu, K., and Presbrey, J. “Tabulator Redux: Writing Into the Semantic Web,” *Proceedings of the 2008 Linked Data on the Web Workshop* (LDOW08), 2008.

[5] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. “Freebase: a collaboratively created graph database for structuring human knowledge,” *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, Vancouver, Canada: ACM, 2008, pp. 1247-1250.

[6] Bollacker, K. Tufts, P., Pierce, T., and Cook, R. “A Platform for Scalable, Collaborative, Structured Information Integration,” *Proceedings of the 6th International Workshop on Information Integration on the We*b, Vancouver, Canada, 2007.

[7] Campanini, S., Castagna, P., and Tazzoli, R. “Platypus Wiki: a Semantic Wiki Wiki Web,” *Proceedings of the 1st Italian Semantic Web Workshop Semantic Web Applications and Perspectives* (SWAP), (Ancona, Italy), 2004, pp. 1-6.

[8] Davies, S., Zeitz, J., and Hatfield, J. “Addressing the Cognitive Difficulties of Expressing N-ary Relations in Semantic Web Data .” (Manuscript submitted for publication.)

[9] Kalyanpur, A., Parsia, B., Sirin, E., Grau, B., and Hendler, J. “Swoop: A web ontology editing browser,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 4, pp. 144-153, 2006.

[10] Kintsch, K. *Comprehension: A Paradigm for Cognition*, Cambridge University Press, 1998.

[11] Krötzsch, M., Vrandecic, D., and Völkel, M. “Semantic MediaWiki,” *Proceedings of the 5th International Semantic Web Conference* (ISWC06), Springer, pp. 935–942, 2006.

[12] Luczak-Rosch, M. and Heese, R. “Linked Data Authoring for Non-Experts.”*Proceedings of the 2009 Linked Data on the Web Workshop* (LDOW09), 2009.

[13] McBride, B. “Jena: A semantic web toolkit,” *IEEE Internet Computing*, vol. 6, pp. 55-59, 2002.

[14] Mitchell, T., Betteridge, J., Carlson, A., Hruschka, E., and Wang, R. “Populating the Semantic Web by Macro-reading Internet Text,” *Proceedings of the 8th International Semantic Web Conference (ISWC09)*, Chantilly, VA: Springer-Verlag, pp. 998-1002, 2009.

[15] Noy, N., Grosso, W., and Musen, M. “Knowledge-acquisition interfaces for domain experts: An empirical evaluation of Protege-2000,” *Proceedings of the 12th Internal Conference on Software and Knowledge Engineering,* Chicago, USA, July, 5-7, 2000.

[16] Pietriga, E. “Isaviz: a visual environment for browsing and authoring rdf models,” *Proceedings of the Eleventh International World Wide Web Conference*, 2002.

[17] Schaffert, S., Eder, J., Grünwald, S., Kurz, T., Radulescu, M., Sint, R., and Stroka, S. “KiWi–A Platform for Semantic Social Software.” *Proceedings of the Workshop on Semantic Wikis*, in conjunction with *The 6th European Semantic Web Conference (ESWC09),* 2009.

[18] Simpson, E. “Measurement of diversity,” *Nature*, vol. 163, 1949, p. 688.

[19] Sowa, J. *Conceptual structures: information processing in mind and machine*, Reading, MA: 1984.

[20] Staab, S., Maedche, A. and Handschuh, S. “Creating metadata for the semantic web: An annotation framework and the human factor,” Technical Report 412, Institute AIFB, University of Karlsruhe, 2001, 2001.

[21] Suchanek, F., Kasneci, G., and Weikum, G., “Yago: a core of semantic knowledge,” *Proceedings of the 16th international conference on the World Wide Web*, p. 706-715, 2007.

[22] Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., and Studer, R. “Semantic Wikipedia,” *Proceedings of the 15th international conference on the World Wide Web*, Edinburgh, Scotland: ACM, pp. 585-594, 2006.

[23] W.A. Woods, “What's in a link: Foundations for semantic networks,” in D. G. Bobrow and A. Collins, eds, *Representation and Understanding*. Academic Press, New York, 1975.