Creating a Standardized Markup Language for Semantic Networks

Brian Harrington and Pia-Ramona Wojtinnek
Department of Computer Science
University of Oxford
Oxford, OX1 3QD, UK
{brian.harrington,pia-ramona.wojtinnek}@cs.ox.ac.uk

Abstract

This paper describes the creation of, and serves as a request for comment on the SemML language for markup of semantic relational information. The major goal of the SemML project is to create a language that can act as an inter-lingua *for a variety of semantic computing applications. In this paper, we discuss the structure of the language, and the ways it which it allows for recursively defined complex concepts, maintains source links which allows for management of multiple world views, and deals with other issues such as temporal data and factual versus non-factual information.*

1. Introduction

Semantic Computing is a rapidly growing field of computer science, combining a variety of subfields of Artificial Intelligence; including Natural Language Processing, Image and Video Analysis, Semantic Web Services and Human Centered Computing among others. With this rapid expansion and integration of multiple diverse topics, there is a strong need for the development of tools and standards that can be used across the field, in order to unify Semantic Computing into a more cohesive area of research. In this paper, we discuss the development of a markup language standard that attempts to facilitate interoperability between semantic technologies by providing an interlingua rooted in the concept of a generalized semantic network.

The main related work is the Resource Description Framework (RDF) [3], which is the standard model for the Semantic Web. Statements in RDF are triples of the form subject-predicate-object while the resources are most commonly identified by URIs. This allows for a description of semantic networks, however, several aspects of networks based on complex facts from different resources cannot be represented as easily and intuitively as desired for semantic computing. In particular, this includes the handling of deep nested structures or reifications, the attribution of temporality and strength to statements as well as annotation of statements with their origin and measures of trust. We present an XML-based language that allows to capture these aspects and we aim to provide a RDF-based syntax in future work.

This document serves as both a guide to the facilities offered by SemML, and an initial request for comment to the Semantic Computing community, to help foster discussion that will guide the future development of the language.

2. SemML - A Semantic Markup Language

The main goal of SemML is to produce an XML based [1] language specifically tailored to the needs of the Semantic Computing community. To this end, the language must be flexible enough to accommodate the wide variety of information types and sources used in the community, while at the same time providing features that will allow, to the extent possible, content rich, unambiguous representation of the information that users might wish to utilize.

We have expressed the semantic network AskNet [2] in SemML and provided an RDF [3] and DOT [4] translation for the part of facilities needed for this type of network [5].

2.1. Network Structure

The primary structure of SemML is *Entity-Relationship* based. Relations are, by default, named and directed, including an *object* and *subject*, but anonymous and undirected relationships are also possible by simply omitting the appropriate fields.

The most basic entity type in SemML is the *primitive*, which represent a single entity or simple concept. Primitives can be assigned entity types, such as *person*, *location* or *organization*, each of which has optional additional attributes. Each primitive also has an optional URI [6], which can be linked to a canonical resource such as Wikipedia, DBpedia [7] or WordNet [8]. This allows unambiguous resolution of the primitives to real world entities.

Relations in SemML are named and directed. Transitive relations have a *subject* and *object*, while intransitive relations will only have a *subject*. It is also possible to have undirected relations or relations with a higher arity through the use of the *link* tag.

Concepts are comprised of relations, along with their associated operands and attributes. Relations can take either primitives or other concepts as their operands. This allows for recursively complex structures and conceptual models, while still maintaining flexibility. As we can see in Figure 1, the *saw* relation can be attached to the concept of *Alice going to London yesterday* just as easily as to a primitive. We can also convey that *Carol* knows all of this information by simply attaching the *knows* relation to the concept encapsulating all of the prior information.

Like primitives, Concepts can be assigned a URI, in order to unambiguously identify them. It is normally assumed that more complex concepts will not make use of the URI, as its purpose will be filled by the relation's *source* tag. However, this has been left as an option to allow reference and identification of simple concepts that are nevertheless non-primitive.

Both relations and concepts can have associated *attributes*. These attributes allow expansion of concepts and relations without having to alter the concepts themselves. Attributes can be viewed as a type of simple, non-transitive relation. In particular, it is possible to create a concept solely out of a primitive or concept and its associated attributes, as is demonstrated in Figure 2.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://www.isc-home.org/SemML"
        xmlns:netml= "http://www.isc-home.org/SemML">

<element name="Example">
        <primitive label="Alice" type="Person" />
        <primitive label="Bob" type="Person" />
        <primitive label="tea" />
        <concept label="GreenTea">
                <attribute label="green" object="tea">green</attribute>
        </concept>
        <concept label="BlackTea">
                <attribute label="black" object="tea">black</attribute>
        </concept>
        <concept label="Alice-Wants-X">
                <relation label="wants1" type="action" temporal="present" strength="1.0">
                        <text>wants</text>
                        <object>Alice</object>
                        <subject>GreenTea</subject>
                        <source>example1</source>
                </relation>
        </concept>
        <concept label="Bob-Wants-X">
                <relation label="wants2" type="action" temporal="present" strength="1.0">
                        <text>wants</text>
                        <object>Bob</object>
                        <subject>BlackTea</subject>
                        <source>example1</source>
                </relation>
        </concept>
</element>
```

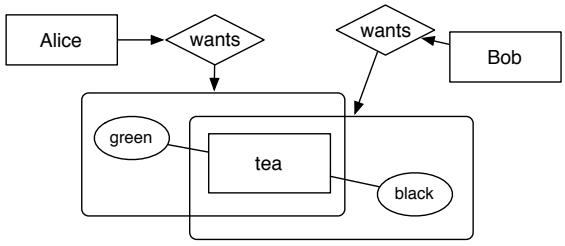

Figure 2: [above] A Simple SemML file and [below] a graphical representation encoding the information "Alice wants green tea. Bob wants black tea."

3. Features of the Language

In this section, we will briefly outline and discuss a number of salient features of SemML that we feel would either a) be of interest and/or benefit to the community; or b) elicit discussion within the community as to the usefulness and implementation details of said feature. As SemML is still in pre-release development, none of these features are "set in stone". However, we feel that the approaches used will allow for adoption by a variety of Semantic Computing projects, while still maintaining room for potential change and improvement in the future.

3.1. Temporality & Strength

Relations in SemML have two attributes that relay their temporal nature. *Temporal*, which can take the values *past*, *present* or *future*, denotes the tense of the relation (which may be different from the grammatical tense of the words used in the label), and *continuity*, which can take the values *incidental*, *temporary* or *permanent*, denotes the duration and scope of the relation.

All relations in SemML also have an associated *strength*. This allows relations of differing salience and import to be represented in the same network. As an example of the use of this feature, if we wanted to add the information "Carol is the mother of Alice" to our diagram in Figure 1, we may decide that this is particularly important to the structure, and thus could give the *mother of* relation a high score value, likewise

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://www.isc-home.org/SemML"
        xmlns:netml= "http://www.isc-home.org/SemML">

<element name="Example">
        <primitive label="Alice" type="Person" />
        <primitive label="Bob" type="Person" />
        <primitive label="Carol" type="Person" />
        <primitive label="London" type="Location" uri="http://en.wikipedia.org/wiki/London" />
        <concept label="Alice-goto-London">
                <relation label="goto1" type="action" temporal="past" strength="1.0">
                        <text>go to</text>
                        <object>Alice</object>
                        <subject>London</subject>
                        <source>example1</source>
                        <attribute label="yesterday1" type="temporal">Yesterday</attribute>
                </relation>
        </concept>
        <concept label="Bob-see-X">
                <relation label="see1" type="action" temporal="past" strength="1.0" continuity="temporary">
                        <text>see</text>
                        <object>Bob</object>
                        <subject>Alice-goto-London</subject>
                        <source>example1</source>

                </relation>
        </concept>
        <concept label="Carol-know-X">
                <relation label="know1" type="action" temporal="current" strength="1.0" continuity="indefinite">
                        <text>know</text>
                        <object>Carol</object>
                        <subject>Bob-see-X</subject>
                        <source>example1</source>
                </relation>
        </concept>
</element>
```

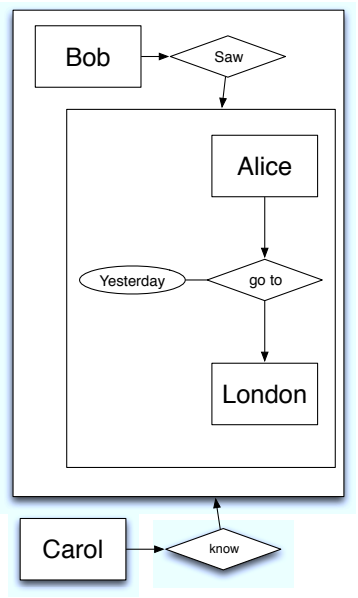

Figure 1: Semantic knowledge structure, including complex concepts made from combinations of atomic concepts and entities, represented by [left] SemML and [right] a semantic network

if we wished to add the information "Kim Jong Il wore a green shirt today" to the information shown in Figure 3, we may decide that this is not very salient information, and therefore give the *wore* relation a lower score.

3.2. Sources & World Views

Not all information is created equal, and not all sources of information can or should be trusted equally. This is, at least in part, the reason that SemML includes *source* structures. Each relation in SemML can be linked to a document that acts as a source for the relation. Currently SemML only supports documents as sources, but it can be easily extended to other media types.

Each document is attached to a source, which may in turn be part of another source. This allows the *trust* variable to be set either at the document or the source level. In the case of several nested sources, the trust is taken from to be the lowest trust score for any nesting level. Likewise, the document trust score overrides the source trust score only in the case where it is lower. This represents the idea that it is possible to get an unreliable document from an otherwise reliable source (perhaps a source could acknowledge that the information is speculative for example), but that one generally only places as much trust into a document as is warranted by its source. Furthermore, this setup allows us to easily manipulate the settings for groups of documents or sources by changing the trust value for their parent source.

This ability to easily manipulate the trust of groups of sources introduces a new feature of SemML, the ability to represent varying world views. By changing the relative trust of various source groups, it is possible to manipulate the SemML structure to represent the knowledge and world view of an individual or group. For example, after encoding information on politics, it would be possible to group sources into liberal and conservative. By manipulating the trust values of these groups, it would be possible to cause the SemML file to represent the world view of an individual who primarily read and believed either the conservative or the liberal sources.

It is also possible, using this source structure, to remove information related to a particular source or group. This can be useful for managing and updating already created files, but moreover, this too can be used to better represent world views. By removing sources to which a group or individual would not have had access, we can better represent their world view, as seen in Figure 3.

The ability to manage sources and thereby manipu-

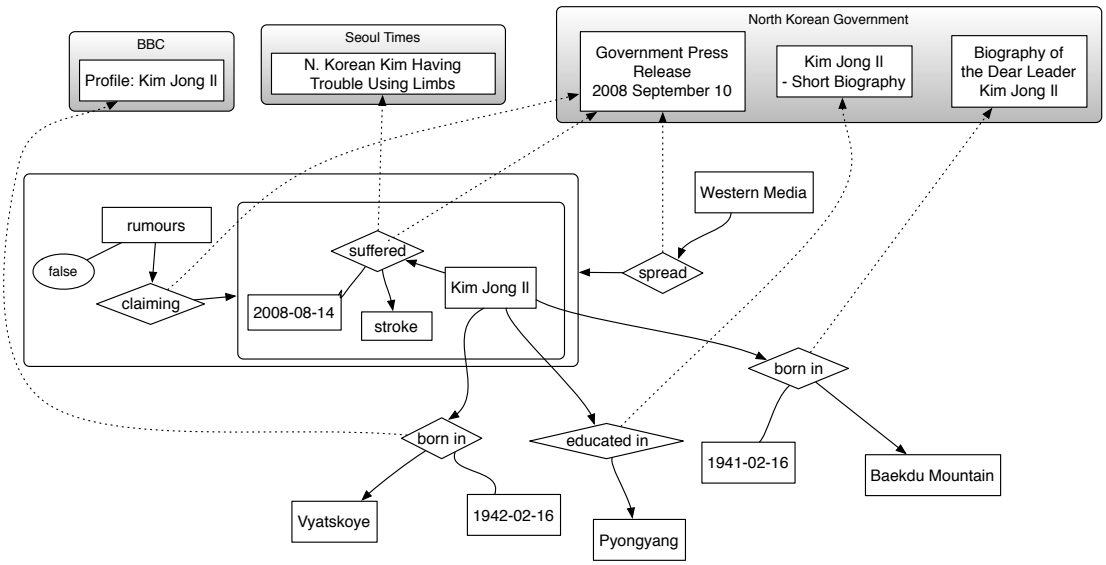

Figure 3: Example semantic structures representing information about Korean Leader Kim Jong Il gathered from a variety of sources. Only North Korean Government sources would be accessible to North Korean citizens.

late world views opens up many interesting avenues of exploration. By constructing a single SemML file, systems can now manage to quickly and non-destructively explore their knowledge base and information as it would be perceived and understood by individuals in various groups with access to or predisposition for only a limited set of information sources.

4. Conclusion and Future Directions

In this paper we have presented and discussed the development of the SemML language. A tool specifically designed to allow members of the Semantic Computing community to define, manipulate and distribute semantic data. While this specification is still in its early stage, it already has the depth and flexibility necessary to make it a valuable tool for sharing data and providing interoperability between systems.

It is important to note that this document is not intended as a full specification of the language, as many details have been necessarily omitted. It should rather be viewed as a point for fostering discussion and debate about the future of Semantic Computing, and the features that this language, and other tools like it will need in order to help the community grow.

References

[1] T. Bray, J. Paoli, E. Maler, F. Yergeau, and C. M. Sperberg-McQueen, "Extensible markup language (XML) 1.0 (fifth edition)," W3C, W3C Recommendation, Nov. 2008, http://www.w3.org/TR/2008/REC-xml-20081126/.

[2] B. Harrington and S. Clark, "Asknet: Automated semantic knowledge network," in *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI'07)*, Vancouver, Canada, 2007, pp. 889–894.

[3] E. Miller and F. Manola, "RDF primer," W3C, W3C Recommendation, Feb. 2004, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[4] E. R. Gansner and S. C. North, "An open graph visualization system and its applications to software engineering," *Software — Practice and Experience*, vol. 30, no. 11, pp. 1203 – 1233, 2000.

[5] P.-R. Wojtinnek, B. Harrington, S. Rudolph, and S. Pulman, "Conceptual knowledge acquisition using automatically generated large-scale semantic networks," in *Proceedings of the 18th International Conference on Conceptual Structures*, 2010.

[6] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," RFC 3986 (Standard), Internet Engineering Task Force, Jan. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3986.txt

[7] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *In 6th Int'l Semantic Web Conference, Busan, Korea*. Springer, 2007, pp. 11–15.

[8] C. Fellbaum, Ed., *WordNet : An Electronic Lexical Database*. Cambridge, Mass, USA: MIT Press, 1998.