A Client Side Approach to Building the Semantic Web

Erik Larson
Digital Media Collaboratory, IC^2 Institute
The University of Texas at Austin
1301 West 25th Street, Suite 300
Austin, TX 78705 USA
512 474 6312 Ext. 265
elarson@icc.utexas.edu

ABSTRACT
In this paper, I describe an alternative approach to building a semantic web that addresses some known challenges to existing attempts. In particular, powerful information extraction techniques are used to identify concepts of interest in Web pages. Identified concepts are then used to semi-automatically construct assertions in a computer-readable markup, reducing manual annotation requirements. It is also envisioned that these semantic assertions will be constructed specifically by communities of users with common interests. The structured knowledge bases created will then contain content that reflects the uses they were designed for, thereby facilitating effective automated reasoning and inference for real-world problems.

Keywords
Semantic web, information extraction, ontology

INTRODUCTION
The World Wide Web is a vast repository of information annotated in a human-readable format. Unfortunately, annotation that is understood by humans is typically poorly understood by machines. Because the Web was designed for human and not machine understanding, facilitating the development of enhancements to the Web such as better search and retrieval, question and answering (Q&A), and automated services through intelligent Web agents is difficult and in many cases not yet practically feasible. Not surprisingly, work on a "next-generation" Web more friendly to machines is already underway, most visibly in the Semantic Web activity championed by Tim Berners-Lee, inventor of the current Web and director of the W3C (http://www.w3c.org), the Web standards and development committee.

The vision of the Semantic Web activity is an "evolution" of the existing Web into one that contains machine-readable markup [Berners-Lee, 2001]. As seen by people, the Semantic Web remains indistinguishable from the current one. Yet machines using the Semantic Web can read Web pages that contain semantic information encoded in a logic-based markup describing their content. This increased power that semantic markup gives to machines also benefits humans: if someone instructs their agent (i.e., their intelligent agent software) to find a bird watching society nearby and to schedule a visit in the next few days, the agent will know that bird watching is a type of outdoor activity and therefore that the weather is a relevant factor, checking the online local forecast (knowing that "nearby" means "local") for signs of thunderstorms, rain, or other kinds or weather incompatible with outdoor activities. The agent can then inform the person of the location of a bird watching society, visiting hours, and good days during the week to go. The evolution of the Web into the Semantic Web, in other words, creates more opportunities for exploiting the rich content on the Web to create value and provide services to everyone.

As laudable as this vision is, there are a number of problems with its practical implementation. First, much of the focus in the current Semantic Web activity is in transforming the HTML content sitting on Web servers to include semantic information in a machine-readable markup language, such as the Resource Description Format (RDF), the DARPA Agent Markup Language with Ontology Inference Layer (DAML+OIL), or the updated version of DAML, the Ontology Web Language (OWL) [RDF, 2003], [DAML, 2003], [OWL, 2003]. This transformation requires, in effect, a re-writing of the billions of pages of content comprising the current World Wide Web—no small feat, particularly since the Semantic Web languages are much less user friendly than simple HTML. True, the Semantic Web markup languages were designed for greater ease of use than traditional knowledge representation (KR) languages based on first-order logic (they are also not as expressive, see [Stevens, R., 2003]), but for non-experts unversed in logic systems, annotating Web pages with RDF, DAML or OWL represents a whole new layer of effort, particularly in relation to the WYSIWYG software for HTML annotation that is now a commonplace.

Second, developers cannot effectively markup Web documents with semantic content unless they understand clearly the *context*—what is the purpose of adding the new information? What function is it serving? What questions will it answer, or services will it provide, that represent a clear benefit in some well-defined context? Without this context, the average Web page developer won't likely see a clear *point* to creating logical markup. Such an effort

would represent, in other words, a purely technical exercise.

Yet a semantic web that facilitates better machine reasoning is indeed desirable and (it is hoped) practically feasible as well. The position taken here is that the "server-side" transformation of Web content in the current Semantic Web activity is, while perhaps helpful in certain cases, nonetheless not a panacea and may even be a hindrance to the task of enhancing the capabilities of the Web for many users. An alternative, "client-side" approach that enables users to effectively transform the existing (HTML) content of the Web into more usable, structured representations that facilitate reasoning within a context of interest will be presented.

THE CLIENT SIDE VISION OF THE SEMANTIC WEB

In a "client-side" semantic web approach, the HTML content of the Web is used "as is." Instead of adding additional markup, a suite of tools and applications are envisioned that extract concepts from Web pages for uploading into a structured knowledge base (KB). The KB is then used for advanced querying, inference, and problem solving.

The client-side approach has a significant advantage over the standard server-side semantic web (hereafter SSSW) because it reduces the content development bottleneck. The client-side semantic web (hereafter CSSW) enables the semi-automatic construction of a "virtual" web on the user's machine (or, in a multi-user environment, on a server that is available to a number of users) that retains hypertext links back to the original Web content but adds a set of logical assertions that captures the meanings germane to the user or users' interests. It therefore helps solve both problems with the SSSW approach explained above: manual annotation effort is reduced by semi-automatic extraction techniques, and because the KB is constructed with a particular interest in mind, there is a clear context for the creation of logical assertions (i.e., the user is creating a KB for a particular purpose that, *ex hypothesi*, is known to the user in advance).

KNOWLEDGE ACQUISITION

Because Web content is left as HTML, the CSSW approach must solve a *knowledge acquisition* problem: how does one transform semi-structured content into structured representations? The short, technical answer to this question is: with an information extraction (IE) system. There are in fact a number of both commercial and open-source IE systems available that can extract concepts and even simple relations from text sources, outputting them into XML or other structured languages (e.g., RDF, DAML). Lockheed Martin's AeroText™ IE system, for instance, can extract key phrases and elements from text documents, as well as perform sophisticated analysis of document structure (identifying tables, lists, and other elements) in addition to complex event extraction and some identification of binary relations [Lockheed Martin Management and Data Systems, 2001-2003].

There are a number of challenges to using information extractions systems for the CSSW. First, no matter how effective an IE system, one cannot yet expect 100% accuracy and recall on arbitrary source documents. This means that false negatives and positives are unavoidable (at least in unconstrained domains). A CSSW system must have functionality in the user interface (UI) to permit selection and editing of extracted results by a human user.

Second, there is a problem of *specificity*: IE systems suitable for handling arbitrary source content (such as, for instance, different Web pages) will not easily support pattern matching for numerous specific concepts. The base functionality of AeroText, for instance, identifies distinctions between 'organizations' and 'people', but not (in the general case) between types of organizations such as the Red Cross (non-profit organization), the University of Texas at Austin (higher education institution), Dell Computer (corporation), and the Smithsonian Institute (art and science institution). A user working on a research project on types of organizations in the United States would get all these types of organizations extracted merely as "Organization"—hardly helpful in this context.

Lastly, there is a problem of identifying proper *relations*: while IE systems excel at identifying patterns for particular things (e.g., proper nouns), they are less effective with relations between things (e.g., binary relations). The reason, to put it bluntly, is that natural language understanding by machines is in too rudimentary a state to handle the grammatical variations in free text occurrences of relations. Extraction rules that do match multi-word patterns and can consistently resolve the semantics of relations embedded in natural language assertions are either domain specific or difficult to construct, or both. (For example, extracting the relation "managed" in "John Doe managed numerous food chains in California before becoming vice president of operations" as an instance of the predicate "managerOf" in an ontology would require distinguishing between this semantics of 'managed' and the following: "Mary managed the sale of half of her stocks before the market took a downturn.")

A solution to the first and second problems above (and to some extent the third) is to *customize* an information extraction system's rule base to perform well on documents containing certain targeted content such as the specific concepts of interest in those documents. This type of solution, however, would not appear to be a complete answer to engineering a CSSW approach, since the original impetus of such an approach was to reduce technical time and effort, but unfortunately customization of IE rule bases is, like manual annotation for the semantic web, a non-trivial technical effort.

However the position advanced here is that the knowledge acquisition challenges specific to the CSSW approach are

nonetheless solvable, either completely or in large degree. It is more difficult to draw the same conclusion of the SSSW. In other words, both approaches have bottlenecks, but the CSSW approach structures the task in such a way that workable remedies seem possible. This suggests that the CSSW approach holds promise for more dynamic progress in the mid, long, and even short term.

THE IMPORTANCE OF CONTEXT

A key difference between the two semantic web approaches is in considerations of *context*. On the one hand, the SSSW requires developers to describe the content of their Web pages in logic, so that the content is understandable (processable) by other software agents with a large range of different goals when visiting Web sites. The problem here is that the developer can't be sure what type of information will be most helpful, and so can't make effective decisions on what to encode. For instance, someone might host a travel site with content on different cities, places, transportation options, fares, special offers, monuments and places of interest. Well, what should they represent logically? Of course it depends on what types of queries and inferences they can expect. It will probably make sense to provide a taxonomy of types:

Car is a type of Vehicle.
Airplane is a type of Vehicle.
Taxi is a type of Car.
Boeing737 is a type of Airplane.

But it is less clear what types of inference rules to spend time supporting: does one anticipate agents and queries that want to check:

((If Place is a Destination and
Customer arrivesAt Destination on Day and
WeatherForecast for Day is Severe) then
Suggest Cancellation or a NewDay)?

Not unreasonable, to be sure. But now creating vocabulary for "WeatherForecast" as well as attributes like "Severe" will be pointless if an agent visiting the site doesn't use such a rule. Given that there might be tens, hundreds, or even thousands of software agents reading travel sites for various reasons (to continue this example), and it is quite likely that there won't be perfect matches between inference rules and logical concepts and assertions on source pages—in which case, nothing will be gained by writing the concepts and assertions—it is hard to make a case for doing the knowledge representation at all.

Now consider the CSSW approach. In this case, we begin with the assumption that a user has a *particular interest* in creating structured content. For instance, a user may want to construct a KB containing assertions about artificial intelligence (AI) research labs in academia and industry, and perform research on whether there are new markets emerging for AI-based techniques. The user can then a) specify the concepts of interest (e.g., research lab, university, corporation, AI techniques, products using AI techniques), b) extract these concepts and upload them into a KB, c) write inference rules that specifically conclude more information of interest from existing information in the KB, such as:

((If ResearchLab hasResearchArea InformationExtraction and ResearchLab hasDirector JohnDoe) then JohnDoe is a ContactInArea-AI),

and finally d) use the KB to ask and answer questions within the context of the research, having now a persistent knowledge source that is focused on a particular domain of interest.

Creating structured content in a context of inquiry also helps reduce information extraction customization requirements. For instance, in a particular context there will typically be a relatively small set of high-value concepts that constitute the main conceptual "framework" of the domain of interest. In the "new market identification" context described above, one might choose, say, the concepts "person", "organization", and "project." An information extraction rule base identifying instances of these generic concepts will require less development time and effort than a corresponding rule base that attempts to match patterns for all subclasses of the generic classes (e.g., subclasses research lab, institution of higher education, and C-corporation for superclass 'Organization'). When the user has an interest in classifying, say, the AI Lab at the University of Texas at Austin as an instance of ResearchLab in the ontology—not just an instance of Organization—this functionality can be handled in the application UI, by providing a means for the user to view, navigate, and modify the ontology and the contents of the KB. The minimal set of 'focused' terms—person, organization, project—provide the pattern matching parameters to the IE system, while any finer-grained classification is handled by the user in the UI.

An alternative approach to "offloading" development effort from IE rule base customization for each specific term of interest to UI based KB classification efforts, is to utilize machine learning (ML) techniques to semi-automatically construct extraction rules for concepts (entities). This approach presents a number of exciting possibilities, most notably the possibility of training an IE system to identify concepts of interest as a user "surfs" the Web. However,

because ML approaches typically require many training examples before accuracy can be achieved (and again, 100% accuracy in unconstrained domains is not likely), such an approach is not a panacea.

For a large KB, training IE rules to find instances for each particular class in an ontology is likely still to be time and effort intensive. However, the approach favored here is to investigate the use of ML techniques for improving the identification of instances of a smaller set of *focused terms* such as explained above, that capture the context of a particular research project. This application of machine learning seems highly promising. For instance, ML techniques could be used to customize an IE rule base to identify research labs as instances of Organization. Users wishing to re-classify research labs as instances of the subclass ResearchLab in the ontology could then perform re-classification by simple specialization of the term in the KB.

SCOPE OF THE CLIENT SIDE APPROACH

The approach outlined above specifically addresses limitations apparent in the SSSW approach. By using IE techniques to semi-automatically extract relevant concepts, and by focusing on a particular research context when undertaking more complicated annotation strategies (e.g., making assertions for automated inference), a usable KB can be constructed that facilitates more advanced Q&A and reasoning in a particular domain.

However there are a number of considerations that should be addressed here. One, because there is still a significant amount of work required to transform free text or HTML markup into a structured, usable KB (some IE rule base customization will be required, as well as manual effort in making relational assertions and classifying concepts in the KB), the CSSW approach will not be suitable for non-persistent "quick" projects that can be answered by performing a few keyword searches on the Web. Such projects are still best handled by existing technologies, such as the Google™ search engine.

Construction of a KB makes the most sense when projects are complex, require the combining of many different types of information, and are relatively long-term and require persistent repositories. In other words, research that spans multiple days, weeks, or even months and that can't easily be handled via conventional browser techniques (saving links into "Favorites" in Internet Explorer) without losing track of the knowledge added and the knowledge still needed, is suited for a more structured approach such as that outlined here. Also, the assumption is that the time involved creating a KB to facilitate reasoning about a particular problem will be offset by the amount of sustained use of the KB a user can expect. Ideally, the KB becomes a semi-permanent repository for a user (or users) that can be referenced, modified, and added to as needed.

Hence, the vision that emerges in the CSSW is a "hybrid" notion of the next generation web, where structured KB's that serve particular purposes co-exist with standard presentational markup, and the choice of whether to enhance the Web is made by particular users within a context of interest.

IMPLEMENTATION

A proof of concept for the CSSW approach is currently under development at Digital Media Collaboratory (DMC), IC^2 Institute, the University of Texas at Austin (http://dmc.ic2.org). The Focused Knowledge Base (FKB) project implements a client server architecture that allows multiple users to login to the system, perform research on the Web, and save facts and knowledge from the Web into a KB. The FKB system uses the AeroText™ information extraction engine to tag 'focused' terms, where they are presented on a separate "knowledge page" in the UI together with a list of relations (taken from the ontology) that can be easily connected to subject and object terms to form a "triple" subject-verb-object assertion in the DAML+OIL language.

Assertions, together with contextual information (e.g., login ID, project name, date, time, area of knowledge) are uploaded into an ontology server. The KAON Ontology server is used to store knowledge in DAML+OIL format [KAON, 2003]. Users can thus browse the Web to identify pages relevant to a research project, enhance the page using AeroText™, add important information not supplied by the IE system (binary relations are presented in drop-down boxes based on the concepts in the subject and object locations), and easily update the KB with the new facts. (Domain specific facts that are uploaded into the KB are subsumed by a top-level ("upper") ontology layer provided by the Standard Upper Merged Ontology (SUMO) [SUMO, 2001].)

In addition to this functionality, DMC is investigating two advanced enhancements to the system. One is the use of an embedded theorem prover. Although DAML+OIL supports standard set-theoretic operations, it provides no facility for constructing rules in the form of logical implications. Such implications, together with a suitable theorem prover such as the JTP theorem prover of Stanford Knowledge Systems Laboratory (http://www.ksl.stanford.edu), make possible the automatic addition of new knowledge (consequences) in the KB from existing knowledge [JTP, 2003]. Rule bases that are focused to add desired information that may be implicit but not noticed in the KB can add significant value. Two, DMC is investigating machine learning approaches to speed construction of IE rule bases suitable for matching instances of focused terms. In particular, relational inductive algorithms for learning information extraction rules such as those designed by Ray Mooney at the University of Texas at Austin (http://www.cs.utexas.edu/users/ml/) show promise especially for Web-based source data [Mooney, 1999].

CONCLUSION

The CSSW is an intriguing alternative to the SSSW vision and ameliorates a number of recognized problems. The high performance of information extraction systems such as AeroText™ coupled with a clearly defined context for Web-based research make the construction of a client-side "virtual" Web with structured repositories of knowledge servicing users and communities of users not just a viable, but an intriguing, option. Further research will include the use of KIF-like rules with DAML+OIL (or OWL) and an embedded theorem prover to generated additional knowledge from existing knowledge. Also, machine learning techniques that work well with Web-based information and can help speed the customization of IE systems are an active area of research that promise to make the CSSW approach even more appealing and feasible as the "next-generation" Web takes shape.

ACKNOWLEDGMENTS

I thank Melinda Jackson and the rest of the DMC staff for providing helpful comments on previous versions of this paper.

REFERENCES

1. [Berners-Lee, 2001] Berners-Lee, T., Hendler, J., Lasilla, O. "The Semantic Web." In Scientific American, May 2001.
2. [DAML, 2003] http://www.daml.org
3. [JTP, 2003] http://www.ksl.stanford.edu/software/JTP/
4. [KAON 2003] http://kaon.semanticweb.org
5. [Lockheed Martin Management and Data Systems, 2001-2003] http://mds.external.lmco.com/mds/products/gims/aero/docs/AeroText-V2.5-Whitepaper-April-2003.pdf
6. [Mooney, 1999] Mooney, R., Califf, M. "Relational Learning of Pattern-Match Rules for Information Extraction" In *Proceedings of the Sixteenth National Conference on Artificial Intelligence* (AAAI-99), Orlando, FL, pp. 328-334, July 1999.
7. [OWL, 2003] http://www.w3.org/2001/sw/WebOnt/
8. [RDF, 2003] http://www.w3.org/RDF/
9. [SUMO, 2001] Niles, I., and Pease, A. 2001. Towards a Standard Upper Ontology. *In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Chris Welty and Barry Smith, eds, Ogunquit, Maine, October 17-19, 2001.
10. [Stevens, R. 2003] Stevens, R., Wroe, C., Bechhofer, S., Lord, P., Rector, A., Goble, C. "Building ontologies in DAML+OIL" In *Comparative and Functional Genomics* Volume: 4, Issue: 1, Date: January/February 2003, Pages: 133-141