

DAML WebScripter: Informal Statement of Work for Mar. 2001 – Mar. 2002

Martin Frank, Pedro Szekely, Bob Neches)
{frank,szekely,rneches}@isi.edu
University of Southern California
Information Sciences Institute

March 14, 2001

1 Summary

The focus of our work is WebScripter, a tool that enables ordinary users to easily and quickly produce live reports that extract and fuse information from multiple heterogeneous sources.

Our effort has always explicitly focused on incentivizing both consumers and producers to use DAML. WebScripter's immediate value for consumers is obvious – they can explore the available DAML data and construct reports combining data from many sources.

WebScripter also has implicit value for DAML producers because the entry cost for using DAML+WebScripter for Web site data is now comparable to using plain XML+XSL — and it has the additional advantages of being able to pull in distributed data and being able to make its data available to the outside world.

Over the next year we want to continue working on the consumer side by improving the power and ease-of-use of WebScripter [section 2.1]. However, the rate at which interesting DAML markup is being produced indicates that a heightened emphasis on producers is also desirable. Insights developed in the past year have suggested some easy and high-payoff ways in which

WebScripter can support and encourage DAML producers by (a) adding a simple Web-based tool for non-technical users to contribute instance data and (b) by making WebScripter an "insidious collaborative ontology translation tool" by explicitly writing out the articulation ontology that every multi-source WebScripter report implicitly represents. This thrust on incentives for DAML producers is described in section 2.2].

Our dual focus on providing incentives both to produce and to consume DAML is pervasive throughout our testbed application. We will continue to deploy DAML applications within our own organization [section 3.1]. Externally, we are investigating the possibility of a connection with Intelink [section 3.2], and otherwise expect to move forward in the area of aviation safety reports [section 3.3]. We are also implementing two joint applications with fellow DAML contractors that focus on showcasing DAML technology itself: a distributed heterogeneous-ontologies DAML application for Information Technology talks in collaboration with UMBC [section 4.1], and one for dynamic DAML tool discovery and invocation with Stanford/Karlsruhe [section 4.2].

As a result of the efforts planned for the upcoming year: (1) the tools will be in place for painless use of DAML – rather than plain XML – for producing Web pages, and (2) mechanisms will be in place through which DAML data can be automatically discovered and combined by mining the DAML articulation ontology output of other people's WebScripter reports. These will combine to provide compelling demonstrations, both technical and applied, of the value of DAML semantic mark-up.

2 Research Directions

This section describes the main thrust of our work.

2.1 Interactive Definition of WebScripter Reports

The core of our effort was, is, and will be easy-to-use interactive DAML data discovery and custom report generation.

We are currently near completion of a tool that given a list of DAML URLs will read that content and display a hierarchical outline of classes on the left. Selecting any class will produce a table on the right which contains

one column per possible attribute and one row per instance; thus the barrier of entry to one's first WebScripter report is significantly lowered.

Below are the next steps in advancing our interactive tools.

1. Let users filter out properties and instances in the above auto-generated reports. [the underlying report generator can already handle this, but users have to edit the textual report specification XML]
2. Let users interactively show the content of multiple classes in a single report. [ditto]
3. Migrate from our current in-memory processing of RDF to a MYSQL database because there is starting to be too much content to process in-memory, even just for the DAML data at ISI. (This is in preparation for the next item.)
4. Implement a first version of the by-demonstration report specification feature described in section D of our original proposal. That is, users will be able to create reports not by navigating ontological classes but rather by typing out examples of the content they are looking for. For example, typing "Pedro", "Martin", and "Baoshi" into cells of a column of a new spreadsheet will make WebScripter guess the ontological type(s) for the content and offer sibling content for the other columns of the report("email", "phone number"). [implementing this well in terms of easy of use and quick response is several months worth of effort by itself]
5. Better support for WebScripter reports that do not naturally correspond to a single table. At the moment, the data structure underlying a WebScripter report is a table which permits multiple values per cell. However, we have found that it is often more natural and easier to create several related tables and then combine them in a post-processing step. For example, the Marine Corps application screen shots consist of one table listing the screen shots and another one listing comments about the screen shots, which we then combine into the final HTML page (using hand-written XSL). We would like to make this an integrated and more end-user accessible feature of WebScripter itself.

We are also working on a low-cost HTML-form-based tool for creating DAML instances (not ontologies) by simply clicking on a link in a WebScripter-based HTML page. This is a smallish but worthwhile effort because we can

then point non-technical users to WebScripser reports and ask them to fill in data for us. For example, we plan to use this to let our Marine Corps users in the CAMERA project add comments and requests to the DAMLized list of screen shots we already have. (We have no intention to provide any interactive tools for DAML *ontologies* as other groups provide some, e.g. Karlsruhe’s OntoEdit and Stanford’s Protege.)

We have deemphasized the idea of building a point-and-click tool for HTML Web pages with embedded DAML, mainly simply because that is not the direction that DAML seems to be headed in – virtually all existing DAML content is in stand-alone files.

2.2 Insidious Collaborative Ontology Translation

One of the unforeseen uses of WebScripser reports is to use them to build articulation ontologies. We plan to enhance our reports to write out the articulation ontology that is inherent in any WebScripser report, in a way so that it can be used to draw inferences about the ontological equivalence of the original source ontologies.

WebScripser already allows users to combine the content of multiple ontologies into a single cell, say “Karlsruhe:Vorname” and “ISI:FirstName”. This is nice because it gets someone’s job done in the here and now but we had previously seen this as a “dead end” effort.

We now believe that there could be great benefit by capturing how these DAML sources were used so that others can benefit from that information. That is we will capture “Captain Miller, working for the Air Force Intelligence Office, considered Karlsruhe:Vorname and ISI:FirstName to be equivalent for his purposes on March 12, 2001”.

This opens up whole new possibilities, such as Amazon-like inferences such as “80% of people who used ISI:FirstName also used Karlsruhe:Vorname” – in effect, it can be used for world-wide insidious¹ collaborative ontology translation.

¹“Insidious” because individual users don’t provide ontology translation information out of altruism but rather produce it as a side effect of their regular job - and they may not even know about this side effect.

3 Application Targets

3.1 ISI

We fully recognize that there is no DAML program credit for using our own tools to organize and process our own content (as we have done for the division people page and the Marine Corps application screen shots).

At the same time, we also recognize that persuading others to use our tools will be difficult if we are not able and willing to use them ourselves. Hence, we will continue to deploy WebScripser DAML applications within our division, and evangelize within ISI and USC.

3.2 Intelink

We are pursuing three technical contributions that WebScripser can make to Intelink: report generation, maintenance of audit trails, and "query mining."

More specifically, the ability of WebScripser to generate multiple alternative views and presentations of an underlying data collection appears to have high potential value for Intelink customers. The mechanisms by which WebScripser reports are constructed make it very easy to keep an audit trail that identifies the sources of information used in the report. WebScripser can present that information either within the report, as a supporting report, or as drill-down into the report. This also has significant value in the intelligence environment.

The process of constructing a WebScripser report gives insight into what information is needed to analysts. By looking at what information is missing after a WebScripser report is constructed, it becomes possible to give information providers feedback about what customers want but are not getting from the materials (e.g., "information in this ontology category was offered by other providers but not by you" or "information about this ontology category was requested but not supplied by anyone"). We call this last process "query mining" because of the important ability it provides for information providers to find out what consumers want to know, as opposed to the onus on consumers to find out what data is available.

These ideas, as part of our notion of insidious ontology translation, were discussed with David Martin-McCormick, and both his current and former bosses, during demonstrations the Friday after the DAML PI Meeting. At their request, follow-up is scheduled later in March.

3.3 Aviation Safety Reports

We are exploring in parallel military interest in the Marine Corps and civilian interest at FAA in modernizing the electronic format of aircraft safety incident reports. These documents cover the nature of each incident, personnel and equipment involved, circumstances, prevailing conditions, and consequences. There is intense interest in improving the ability to analyze these records in order to understand patterns and causal linkages. WebScripser contributes by providing a tool to create complex queries to mine the data.

As part of this application (as well as Intelink and current applications at PACOM), we are still pursuing connections to GeoWorlds, focusing upon (1) leveraging that system to display map-based reports in addition to tabular displays; (2) developing a complementary relationship between GeoWorlds search capabilities for unmarked materials and WebScripser's interface to DAML-ized materials; and (3) using GeoWorlds exiting HTML distilling capabilities to generate DAML mark-up for plain HTML pages.

4 Collaboration with other DAML Contractors

This appendix lists our planned collaboration with other groups. There is some vagueness to the actual amount of effort that we will expend to work with other groups; this is intentional because work in direct support of a government application (such as Intelink) will likely supercede and supplant some of the effort we would otherwise spend with other DAML contractors.

4.1 Tim Finin's Group (UMBC, JHU)

contact: Tim Finin, finin@umbc.edu

We will create new computer science talks DAML data (for ISI and the USC Computer Science Department) according to our own ontology.

We will then exercise WebScripser's capabilities to not just create combined reports of the UMBC and ISI talks data, but also to write out ontology translation DAML statements.

This way, e.g. a third university listing their IT talks and others can do so by building on the ontological-equivalence DAML statements about UMBC and ISI written out by previous WebScripser reports.

This could provide the nucleus for other Computer Science departments to participate in DAML-based heterogeneous-ontology exchange of CS talks information.

4.2 Karlsruhe (subcontractor to the Stanford DB Group)

contact: Siegfried Handschuh (handschuh@acm.org), Stefan Decker (stefan@db.stanford.edu), Rudi Studer, Gio Wiederhold

Karlsruhe and ISI would like to cooperate on building simple initial heterogeneous DAML service ontologies that can eventually be used to dynamically discover and invoke DAML viewers on the Web. Among other things, this effort will ...

1. lead to an available and self-updating list of available DAML tools and services at Stefan Decker's semanticweb.org
2. make Karlsruhe's and ISI's existing Java-based DAML viewers available as services on the Web
3. exercise WebScripser's new ontology translation capabilities
4. achieve some initial semantic interoperability of services that advertise their capabilities via heterogeneous-ontologies DAML

A Detailed Work Plan with Karlsruhe/Stanford DB group

This appendix elaborates on 4.2. (We are still working on producing a more detailed work plan with UMBC; its absence in this report should not be construed as us preferring one collaborator over another — we see each as equally important.)

The phasing we envision with Karlsruhe/Stanford is roughly as follows:

1. Create a “shallow” upper ontology of currently available DAML tools (parsers, crawlers, viewers, editors). [Karlsruhe]

2. Classify currently available tools according to that ontology. [all]
3. Create polished WebScripter reports of the available tools. [ISI]
4. Put up this living (self-refreshing) list of DAML tools on semanticweb.org [Stefan Decker, support from ISI]
5. Create a dedicated ontology for a Web-based DAML viewer (a servlet that when given a list of URLs will load the DAML and render it to HTML in some fashion) [both Karlsruhe and ISI, independently, separately conceived ontologies, separately written viewing service].²
6. Use WebScripter to map each ontology into the semanticweb.org ontology (as a result, the viewing services show up on Stefan's Web page). [ISI and Karlsruhe]
7. Use WebScripter to produce DAML that asserts the equivalence of invocation parameters of the two viewing services. [ISI]
8. Write a program that takes a DAML database, dynamically discovers available viewers by consulting WebScripter ontology equivalence information, and then actually invokes one of the viewing services. [Karlsruhe and ISI]

This final step presents some first semantic interoperability at the tool level (new viewing services can be discovered and incorporated without having to re-write any code of the invoking program). Our effort intentionally stops short of fully-automatic ontology translation (there are other DAML contractors funded to do that); instead, in at least one WebScripter report somewhere in the world someone must have specified the mapping of a new viewing ontology to another one (that in turn via transitive closure can eventually be mapped into the viewing service ontology that the program of the last step above internally uses).

²Java-based viewing services already exist on both sides – they just need to be hooked up to a servlet. In particular, Karlsruhe has a graph-based viewer that is well suited for ontology-centric DAML; while ISI has tabular views that are well suited for instance-centric DAML.