WebScripter: World-Wide Grass-roots Ontology Translation via Implicit End-User Alignment

Research Paper Category

Martin Frank, Pedro Szekely, Robert Neches, Baoshi Yan, Juan Lopez
Distributed Scalable Systems Division
Information Sciences Institute
University of Southern California
{frank,szekely,rneches,baoshi,juan}@isi.edu

ABSTRACT

Ontologies define hierarchies of classes and attributes; they are meta-data: data about data. XML Schema and RDF Schema are both (lightweight) ontology definition languages in that sense. In the "traditional" approach to ontology engineering, experts add new data by carefully analyzing others' ontologies and fitting their new concepts into the existing hierarchy. In the emerging "Semantic Web" approach to ontology engineering, ordinary users may not look at anyone's ontology before creating theirs – instead, they may simply define a new local schema from scratch that addresses their immediate needs, without worrying how their data may some day integrate with others'.

This paper describes an approach and implemented system for translating between the countless mini-ontologies that the Semantic Web approach yields. In this approach, ordinary users graphically align data from multiple sources in a simple spreadsheet-like view without having to know anything about ontologies or even taxonomies. The resulting web of equivalency statements can then be mined to help other users find related ontologies and data, and to automatically align the data with theirs.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: User/Machine Systems—*Human information processing*; H.3.3 [**Information Systems**]: Information Search and Retrieval—*Information filtering, Relevance feedback*; H.3.5 [**Information Systems**]: Online Information Services—*Data sharing, Web-based services*

General Terms

Collaborative filtering, recommender systems, social information filtering, ontology alignment, ontology translation

Keywords

Meta-data, DAML, RDF Schema, RDF, XML Schema

1. INTRODUCTION

Imagine that you work for an emergency preparedness agency and that you were just handed the job of constructing and maintaining a list of public health experts employed by U.S. universities.

Doing this manually on the (non-semantic) Web would be a monumental effort, both in terms of the initial effort and in the continuous effort to keep the list up to date. The only options are to either do the job completely manually in a text file or spreadsheet (quickly outdated), or to write wrapper software specific for each university's Web pages that extracts the experts (the wrappers constantly break as universities change their Web page designs).

Now let us presume that all universities list their personnel in a Semantic Web [10] format, such as RDF Schema [1]. This improves on the current sitation (because you don't have to work instance by instance but rather concept by concept) but your job is still rather monumental because the sources will likely use a myriad of different ontologies.

We have a vision and partial implementation addressing this problem by (a) making it easy for individual users to graphically align the attributes of two separate externally-defined concepts, and (b) making it easy to re-use the alignment work of others.

2. OVERVIEW

Figure 1 depicts a number of home pages, marked up with DAML information about the authors, located somewhere in the world. The DAML instances in these Web pages are organized according to one or more ontologies, such as an ISI ontology of people, a Stanford ontology of people, and a Karlsruhe ontology of people. The challenge is to produce a report incorporating all of that information with minimal effort.

At a high level, the WebScripter concept is that users extract content from apparently ordinary Web pages and paste that content into what looks like an ordinary spreadsheet (lower left corner of Figure 1).

What users implicitly do in WebScripter - without expending extra effort - is to build up an articulation ontology containing equivalency statements. For example, this artic-

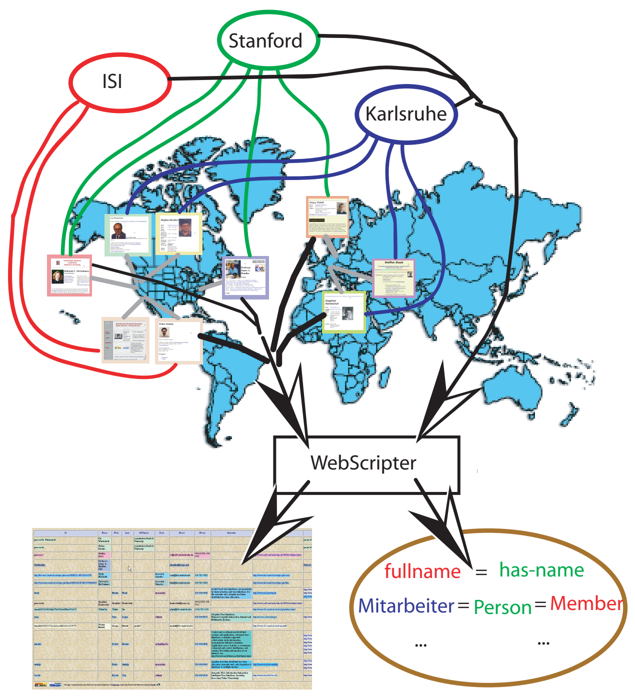
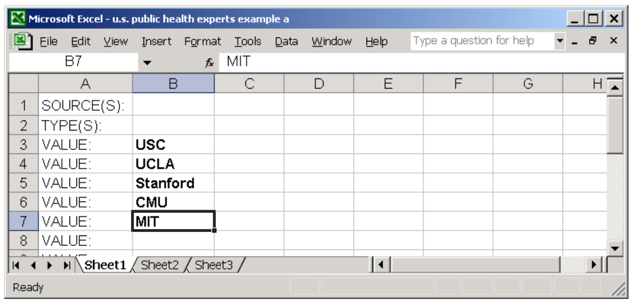


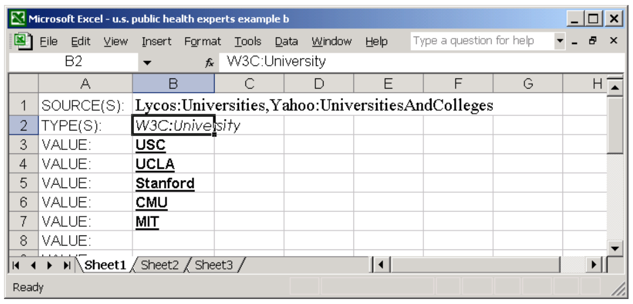
Figure 1: Working With Multiple Data Sources In Multiple Ontologies.


Microsoft Excel - u.s. public health experts example a
File Edit View Insert Format Tools Data Window Help
Type a question for help
Sheet1 Sheet2 Sheet3
Ready

B7 MIT

	A	B	C	D	E	F	G	H
1	SOURCE(S):							
2	TYPE(S):							
3	VALUE:	USC						
4	VALUE:	UCLA						
5	VALUE:	Stanford						
6	VALUE:	CMU						
7	VALUE:	MIT						
8	VALUE:							

Figure 2: Visionary Example: The user types as-yet-unrecognized example values.


Microsoft Excel - u.s. public health experts example b
File Edit View Insert Format Tools Data Window Help
Type a question for help
Sheet1 Sheet2 Sheet3
Ready

B2 W3C:University

	A	B	C	D	E	F	G	H
1	SOURCE(S):	Lycos:Universities,Yahoo:UniversitiesAndColleges						
2	TYPE(S):	*W3C:University*						
3	VALUE:	USC						
4	VALUE:	UCLA						
5	VALUE:	Stanford						
6	VALUE:	CMU						
7	VALUE:	MIT						
8	VALUE:							

Figure 3: Visionary Example: The system determines data sources and a classification.

ulation ontology expresses that the attribute that ISI calls “fullname” is the same as the one Stanford calls “has-name”; and that the object Karlsruhe calls “Mitarbeiter” Stanford calls “Person” and ISI calls “Member” are the same for the purposes of this report (lower right corner of Figure 1).

We believe that in the long run, this articulation ontology will be more valuable than the data the users happened to obtain when they constructed the original report. Its equivalency information reduces the amount of work future WebScripter users have to perform when working in the same domain.[1] Thus, in some sense, you don’t just use the Semantic Web when you use WebScripter, you help build it as you go along.

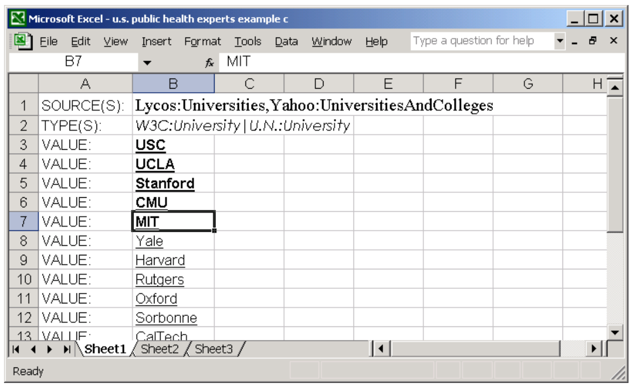
3. VISIONARY EXAMPLE

This section presents a detailed step-by-step vision of what WebScripter (and a future Semantic Web) could be; we will later present a step-by-step example of what our current implementation can already do (with existing RDF(S) data on the Web produced by others). In this example, the application is to quickly produce a self-updating list of faculty at U.S. universities that are public health experts, listing their specialization. The user starts WebScripter and types the names of several universities into the first column. At the point shown in Figure 2, truly nothing is known about these hand-typed values.

After the user selects “classify” from a menu, WebScripter uses a list of well-known indices to find an existing taxonomy that matches all of the typed phrases (note that commercial search engines do not have to be DAMLized to be useful to our reasoning here). Yahoo:UniversitiesAndColleges and Lycos:Universities both apply. The universities now appear underlined because they are recognized by the system - double-clicking on them brings up their web pages. The system then fetches their DAML-enabled Web pages in the background, and computes a minimal covering set of declared DAML IS-A types that cover all the universities. In our example, all of the current universities declare to be instances of the World-Wide Web Consortium’s W3C:University concept (Figure 3).

The user now selects “find more” from the menu bar. The system will fetch every entity that the two known indices point to (several hundred). It simultaneously performs a different type of analysis: Which are the RDF(S) subclass-of types that are declared by more than 10% of the entities (result: U.N.:University, UsPostalService:Recipient, W3C:University, and IRS:NonProfitInstitution) [this is a recall test]? Of these, which apply to less than 1% of nearby categories of the same index (remaining result: W3C:University and U.N.:University) [this is a precision test]. The latter one is now automatically treated as an alternatively valid type, and WebScripter will include every entity declaring to be one of these types, thereby finding institutions not yet listed by the well-known indices. Note that there are no duplicate universities in this column (such as “UCLA” and “UC Los Angeles”). The challenge, of course, is to be able to determine that they are “different”, as they subscribe to different DAML ontologies. One possiblity is that any Semantic Web description of an entity existing on the Web contain its normalized HTTP URL in a standardized attribute, which can serve as a simple unique id for comparisons across ontologies (first choice for disambiguation in our example). Another possibility is that they contain

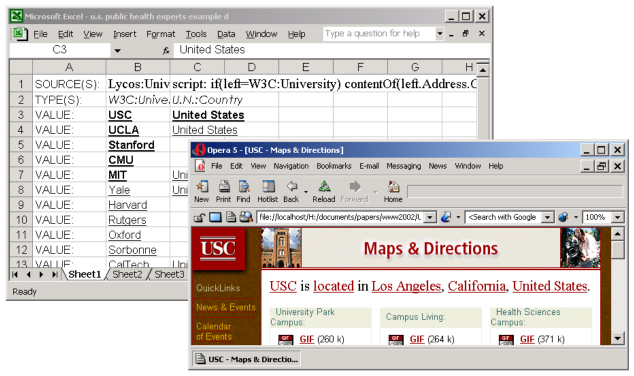
[1]Who benefits depends on your willingness to share that information, of course - it could be the world, your organization, your workgroup, or just yourself.


Microsoft Excel - u.s. public health experts example c
File Edit View Insert Format Tools Data Window Help
Type a question for help
Sheet1 Sheet2 Sheet3
Ready

B7 MIT

	A	B	C	D	E	F	G	H
1	SOURCE(S):	Lycos:Universities,Yahoo:UniversitiesAndColleges						
2	TYPE(S):	*W3C:University | U.N.:University*						
3	VALUE:	**USC**						
4	VALUE:	**UCLA**						
5	VALUE:	**Stanford**						
6	VALUE:	**CMU**						
7	VALUE:	**MIT**						
8	VALUE:	Yale						
9	VALUE:	Harvard						
10	VALUE:	Rutgers						
11	VALUE:	Oxford						
12	VALUE:	Sorbonne						
13	VALUE:	CalTech						

Figure 4: Visionary Example: The system auto-completes the user-provided values.


Microsoft Excel - u.s. public health experts example d
File Edit View Insert Format Tools Data Window Help
Type a question for help
Sheet1 Sheet2 Sheet3
Ready
Opera 5 - [USC - Maps & Directions]
File Edit View Navigation Bookmarks E-mail Messaging News Window Help
New Print Find Hotlist Back Reload Forward Home
file://localhost/H:/documents/papers/www2002/L
<Search with Google
100%
USC
Maps & Directions
QuickLinks
News & Events
Calendar of Events
University Park Campus:
Campus Living:
Health Sciences Campus:
GIF (260 k)
GIF (264 k)
GIF (371 k)
USC - Maps & Directio...

C3 United States

	A	B	C	D	E	F	G	H
1	SOURCE(S):	Lycos:Univ	script: if(left=W3C:University) contentOf(left.Address.C					
2	TYPE(S):	*W3C:Unive*	*U.N.:Country*					
3	VALUE:	**USC**	**United States**					
4	VALUE:	**UCLA**	United States					
5	VALUE:	**Stanford**						
6	VALUE:	**CMU**						
7	VALUE:	**MIT**	Uni					
8	VALUE:	Yale	Uni					
9	VALUE:	Harvard						
10	VALUE:	Rutgers						
11	VALUE:	Oxford						
12	VALUE:	Sorbonne						
13	VALUE:	CalTech	Uni					

USC is located in Los Angeles, California, United States.

Figure 5: Visionary Example: Combining HTML navigation with embedded DAML semantics.

(possibly composite) keys that point into popular external ontologies, for the same reason ("companies in this ontology are uniquely identified by their UsTreas:IRS:TaxPayerId"), ("universities are identical if they point to the same UsPostalService:UsStreetAddress"). This gets the user to the state of Figure 4.

In this vision of a future Semantic Web, the user has to know little to get a lot of leverage out of the existing semantic information: (1) The user did not have to do anything but type out some university names that came to mind – he or she didn't have to understand an ontological query language or the notion of an ontology or even a taxonomy for that matter - yet the result is perfectly ontologically typed. (2) Very little DAML has to be in place for this to work: for this particular example, two external DAML ontologies of existing non-DAML university web sites should be sufficient for the inferencing of this example. (3) Data from two different ontologies can be seamlessly integrated without the need for pre-merging/translation between the ontologies.

In this example, the user now demonstrates that she wants to extract the nationality of the universities, in the following manner (Figure 5). She double-clicks on USC, which brings up a Web browser to the (hypothetically) DAML-enabled USC home page. The user then clicks on "Maps & Directions", and copies and pastes "United States" from that page, which is not just plain text but carries its embedded DAML type.[2]

In response, the system now fills in all those cells that use the same underlying W3C university definition, by inferring the ontological path from university to country and applying it to all other instances of this ontology. In our particular case, the user is best served by now doing the same for the UN-based university entry "Stanford" (not shown) because there are only two ontologies involved.[3] As a result, all missing countries in the second column are now filled in. The user selects a United States cell in the second column and invokes "filter by" from the right-click menu, checks "United States", and clicks OK, which removes Oxford and all other foreign-university rows. Performing a number of substantially similar steps, the user can navigate to the universities' chemistry, biology, and medical departments, from the department to the faculty, from the faculty to their research interests, and filter by a particular research area, resulting in the table shown in Figure 6. (As before, bold entries were provided or demonstrated by the user; but we are no longer underlining recognized cells below for readability).

In the end, what users want is a report containing the information that serves their immediate needs. In our approach, users build a report in steps, by manipulating the data it contains so far to refine it and to add more. This is a qualitatively easier task than working with a query, which is an inherently more abstract specification. In our approach, a final report may contain dozens or hundreds of single-step scripts that operate on DAML markup. The equivalent query could be enormously complicated (perhaps several pages long), but users never have to see it with this approach.

Now that this hypothetical WebScripter report is defined, its data can be refreshed at any time, and it itself can become the source for further Web scripting as it carries all its DAML within the generated HTML report.

4. IMPLEMENTED EXAMPLE

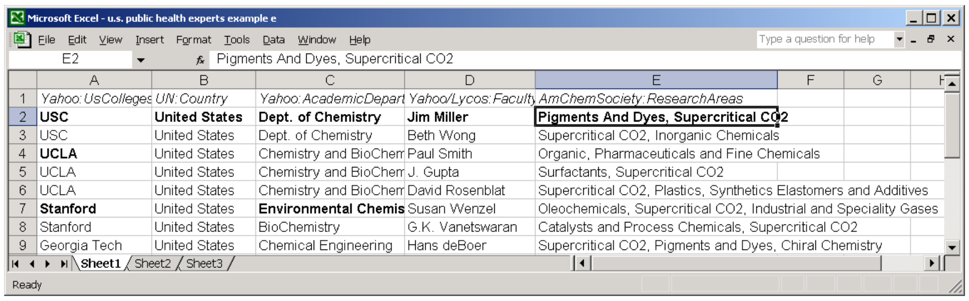
In our initial implementation we have focused on making it easy for ordinary (non-programming, never heard of ontologies) users to contruct reports from multi-ontology DAML data. This section first describes a step-by-step walkthrough of using WebScripter as implemented to combine DAML personnel data from different organizations on the Web. It then describes how the resulting implicit ontology alignment data benefits other users in constructing similar reports.

4.1 Constructing a first report from scratch

Imagine that you work for the government DAML program office, and that your job is to maintain a list of personnel funded by that program, and let's assume that all of the contractors provide their personnel data in some DAML format. The first task is to find the URLs where the vari-

[2]Note that one would not have to internally instrument a Web browser to achieve this level of integration – one could know which page the user is looking at through a proxy Web server and receive the copied HTML+DAML out of the window system's paste buffer.

[3]If there is more than two WebScripter could attempt to produce a generalized "fuzzy" script that will work for all remaining university ontologies given two (or more) examples ("extract the attribute whose name contains Country or Nation in the top-level concept or in a sub-concept called Location or Address").


Microsoft Excel - u.s. public health experts example e
File Edit View Insert Format Tools Data Window Help
Type a question for help
Sheet1
Sheet2
Sheet3
Ready

E2 | Pigments And Dyes, Supercritical CO2

	A	B	C	D	E	F	G
1	*Yahoo:UsColleges*	*UN:Country*	*Yahoo:AcademicDepart*	*Yahoo/Lycos:Faculty*	*AmChemSociety:ResearchAreas*		
2	**USC**	**United States**	**Dept. of Chemistry**	**Jim Miller**	**Pigments And Dyes, Supercritical CO2**		
3	USC	United States	Dept. of Chemistry	Beth Wong	Supercritical CO2, Inorganic Chemicals		
4	**UCLA**	United States	Chemistry and BioChem	Paul Smith	Organic, Pharmaceuticals and Fine Chemicals		
5	UCLA	United States	Chemistry and BioChem	J. Gupta	Surfactants, Supercritical CO2		
6	UCLA	United States	Chemistry and BioChem	David Rosenblat	Supercritical CO2, Plastics, Synthetics Elastomers and Additives		
7	**Stanford**	United States	**Environmental Chemis**	Susan Wenzel	Oleochemicals, Supercritical CO2, Industrial and Speciality Gases		
8	Stanford	United States	BioChemistry	G.K. Vanetswaran	Catalysts and Process Chemicals, Supercritical CO2		
9	Georgia Tech	United States	Chemical Engineering	Hans deBoer	Supercritical CO2, Pigments and Dyes, Chiral Chemistry		

Figure 6: Visionary Example: The end result in this fictious example.

ous DAML resides. BBN's crawled ontology library comes closest to a Yahoo-style portal for DAML content [2]. This site contains a registry for DAML content root files, which a crawler uses as starting points to find more DAML files. Teknowledge built a DAML search engine for that ontology library which is a good starting place for finding DAML content [3].

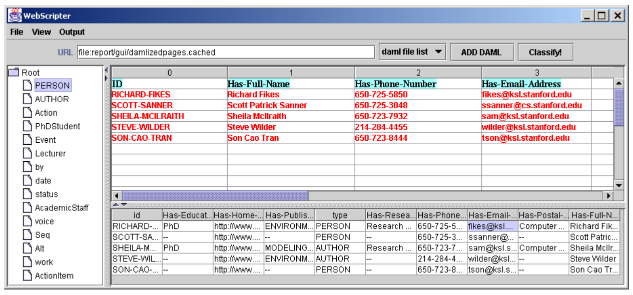
In this example, the DAML sources can be found by querying the Teknowledge search engine for the terms "Person", "Employee", and "Staff" (which will return a large number of hits of non-DAML contractors), or alternatively it can be found by collecting the regular project Web pages and personal home pages of the DAML contractors (because they embed DAML content inside the HTML pages).

For the sake of this example, we started WebScripter and loaded DAML from just the Stanford Database and Knowledge Systems groups by copying and pasting the URLs of their DAML pages into WebScripter's "Add DAML" dialog box. WebScripter then displays the class hierarchy of that DAML, intermixing the concepts from the two separate ontologies. The user can browse the content by selecting classes, which displays all of their (local and inherited) attributes as columns, and their data instances as rows.

In this example, we started a new report by (1) choosing "New Report" from a menu, (2) selecting Person in the class hierarchy, and (3) selecting three columns of Person to include in the report. The latter is done by selecting a cell in the data display for Person and choosing "Add as new column" from the right-click menu, once each for the Has-Full-Name, Has-Phone-Number, and Has-Email-Address columns. The resulting WebScripter display is shown in Figure 7. (Note that the first of the four columns, the DAML instance identifier column, was automatically inserted when the first column was added to the report. The column is hidden from the generated report Web page by default.)

In this example, we will now add and align data from a different research group using a different ontology. This is done by (1) selecting PhDStudent in the class hierarchy to display its instance data, (2) selecting a cell in the "name" column of that instance data and choosing "Add to column 1" from the right click menu, and (3) repeating the second step for the "phone" and "email" columns. Figure 8 shows the combined data from the two groups.

This in a nutshell is how WebScripter looks to the users. This report can then be published in various formats, including as a plain Web page that color-codes its content based on where it came from; Figure 9 shows a snapshot of a large DAML personnel report that loads data from more than 30

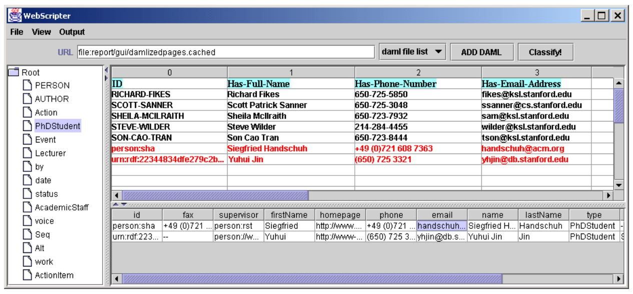


URL file:report/gui/damlizedpages.cached

0	1	2	3
ID	Has-Full-Name	Has-Phone-Number	Has-Email-Address
RICHARD-FIKES	Richard Fikes	650-725-5850	fikes@ksl.stanford.edu
SCOTT-SANNER	Scott Patrick Sanner	650-725-3048	ssanner@cs.stanford.edu
SHEILA-MCILRAITH	Sheila McIlraith	650-723-7932	sam@ksl.stanford.edu
STEVE-WILDER	Steve Wilder	214-284-4455	wilder@ksl.stanford.edu
SON-CAO-TRAN	Son Cao Tran	650-723-8444	tson@ksl.stanford.edu

id	Has-Educat...	Has-Home-...	Has-Publis...	type	Has-Resea...	Has-Phone...	Has-Email-...	Has-Postal-...	Has-Full-N...
RICHARD-...	PhD	http://www....	ENVIRONM...	PERSON	Research ...	650-725-5...	fikes@ksl....	Computer ...	Richard Fik...
SCOTT-SA...	--	http://www....	--	PERSON	--	650-725-3...	ssanner@...	--	Scott Patric...
SHEILA-M...	PhD	http://www....	MODELING...	AUTHOR	Research ...	650-723-7...	sam@ksl.s...	Computer ...	Sheila McIlr...
STEVE-WIL...	--	http://www....	ENVIRONM...	AUTHOR	--	214-284-4...	wilder@ksl...	--	Steve Wilder
SON-CAO-...	--	http://www....	--	PERSON	--	650-723-8...	tson@ksl.s...	--	Son Cao Tr...

Figure 7: Implemented Example: Initial report of Stanford KSL personnel.



URL file:report/gui/damlizedpages.cached

0	1	2	3
ID	Has-Full-Name	Has-Phone-Number	Has-Email-Address
RICHARD-FIKES	Richard Fikes	650-725-5850	fikes@ksl.stanford.edu
SCOTT-SANNER	Scott Patrick Sanner	650-725-3048	ssanner@cs.stanford.edu
SHEILA-MCILRAITH	Sheila McIlraith	650-723-7932	sam@ksl.stanford.edu
STEVE-WILDER	Steve Wilder	214-284-4455	wilder@ksl.stanford.edu
SON-CAO-TRAN	Son Cao Tran	650-723-8444	tson@ksl.stanford.edu
person:sha	Siegfried Handschuh	+49 (0)721 608 7363	handschuh@acm.org
urn:rdf:22344834dfe279c2b...	Yuhui Jin	(650) 725 3321	yhjin@db.stanford.edu

id	fax	supervisor	firstName	homepage	phone	email	name	lastName	type
person:sha	+49 (0)721 ...	person:rst	Siegfried	http://www....	+49 (0)721 ...	handschuh...	Siegfried H...	Handschuh	PhDStudent
urn:rdf:223...	--	person://w...	Yuhui	http://www-...	(650) 725 3...	yhjin@db.s...	Yuhui Jin	Jin	PhDStudent

Figure 8: Implemented Example: Adding and aligning Stanford Database personnel.

different sources [4]. The largest such report we have generated is 3.3MB, taking 8.7MB of DAML input, and running for about 45 seconds.

The Web page embeds the WebScripter report definition, thus it can be re-run at any time and will then possibly show more people (presuming their DAMLized Web pages can be found by following just one link from the two group Web pages, and presuming their DAML instance data follows one of the two ontologies).

There are a large number of WebScripter features that we will not discuss here – such as un-loading DAML sources, deleting columns, re-arranging columns, filtering rows, and sorting by multiple criteria, and so on – because they are what you would expect from any DAML report generator. Instead, we'll focus on the generated DAML equivalency statements shown in Table 1.

These statements can be automatically published on a Web site and registered as a new DAML content root in BBN's DAML content library. Consequently, you can then

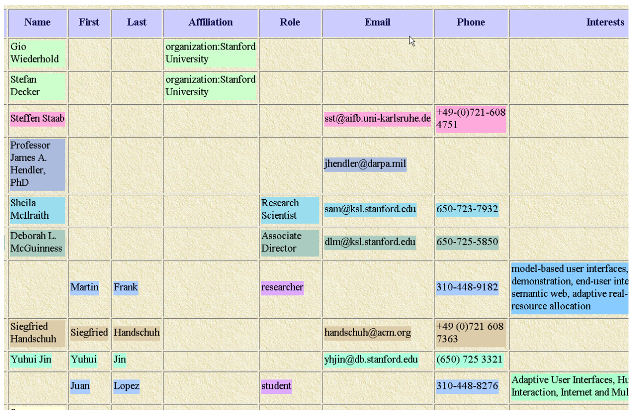
```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdf="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:daml="http://www.daml.org/2001/03/daml+oil#">

<rdfs:Class
  rdf:about="http://ksl.stanford.edu/Projects/DAML/ksl-daml-desc.daml#PERSON">
  <daml:sameClassAs rdf:resource=
  "http://www.semanticweb.org/ontologies/swrc-onto-2000-09-10.daml#PhDStudent"/>
</rdfs:Class>

<rdfs:Property
  rdf:about="http://ksl.stanford.edu/Projects/DAML/ksl-daml-desc.daml#Has-Full-Name>
  <daml:samePropertyAs rdf:resource=
  "http://www.semanticweb.org/ontologies/swrc-onto-2000-09-10.daml#name"/>
</rdfs:Property>
<rdfs:Property
  rdf:about="http://ksl.stanford.edu/Projects/DAML/ksl-daml-desc.daml#Has-Phone-Number>
  <daml:samePropertyAs rdf:resource=
  "http://www.semanticweb.org/ontologies/swrc-onto-2000-09-10.daml#phone"/>
</rdfs:Property>
<rdfs:Property
  rdf:about="http://ksl.stanford.edu/Projects/DAML/ksl-daml-desc.daml#Has-Email-Address>
  <daml:samePropertyAs rdf:resource=
  "http://www.semanticweb.org/ontologies/swrc-onto-2000-09-10.daml#email"/>
</rdfs:Property>

</rdf:RDF>
```

Table 1: Implemented Example: Resulting DAML equivalency statements.



Name	First	Last	Affiliation	Role	Email	Phone	Interests
Gio Wiederhold			organization:Stanford University				
Stefan Decker			organization:Stanford University				
Steffen Staab					sst@aifb.uni-karlsruhe.de	+49-(0)721-608 4751	
Professor James A. Hendler, PhD					jhendler@darpa.mil		
Sheila McIlraith				Research Scientist	sam@ksl.stanford.edu	650-723-7932	
Deborah L. McGuinness				Associate Director	dlm@ksl.stanford.edu	650-725-5850	
	Martin	Frank		researcher		310-448-9182	model-based user interfaces, demonstration, end-user inte semantic web, adaptive real- resource allocation
Siegfried Handschuh	Siegfried	Handschuh			handschuh@acm.org	+49 (0)721 608 7363	
Yuhui Jin	Yuhui	Jin			yhjin@db.stanford.edu	(650) 725 3321	
	Juan	Lopez		student		310-448-8276	Adaptive User Interfaces, Hu Interaction, Internet and Mul

Figure 9: Snapshot of (a fragment of) a large-size WebScripter DAML people report.

make use of the equivalency statements by selecting the "Extended with Equivalence" option in Teknowledge's DAML search engine (note that it can take up to 24 hours for the statements to make it into BBN's cache and then up to another week from there into Teknowledge's search engine cache). Concretely, if you for example now query for all instances of person ("?x type Person") in the first ontology in that fashion you will now also retrieve PhDStudent instances from the second ontology.

4.2 Constructing a second report using the alignment data

We have also implemented an intial use of the WebScripter-generated equivalency statements in WebScripter itself: if you start it with the *insert-equivalents* flag it will automatically add and align any classes that it has *sameClassAs* and *sameInstanceAs* data for. It reads these equivalency statements from a fixed location on our Web site to which you can contribute more via the "Easy Publish" menu in WebScripter.

Let's assume that a second user comes along later whose job it is to maintain a list of researchers with Semantic Web expertise, plus their email addresses and home pages. She starts WebScripter in the same way as above, selects for example PhDStudent and adds "name" as the first column in her report. At that point, WebScripter will not only add all instances of Person, but also automatically align their names into the column. Similarly, when then selecting the email address for either Person or PhDStudent and saying "Add as new column" WebScripter will fill in the email addresses for the other ontology as well. This will not happen after she adds *Has-Home-Page* as a new column (as there is no existing equivalency data) so that she has to manually select *homepage* and say "Add to column". (However, if she is willing to share her alignment data via the "Easy Publish" option future users do not have to align this column by hand either.)

5. THOUGHTS ON INCENTIVIZING PRODUCERS

As of the time of writing, one issue we encountered is that there is not that much interesting, continuously updated RDF(S), much less DAML, available on the Web today.[4] What made the original Web take off was that there was an immediate incentive for producers to use the technology because it was an easy way to publish information. We currently see no strong motivation for producers to put work into putting out RDF(S) in addition to their regular HTML pages, but there is at least a compelling intra-organizational benefit in using RDF(S) and WebScripter to generate regular HTML pages by pulling RDF from various pages within the organization.

To be more concrete, once a DAML-enabled document is published on the Web, WebScripter makes it easy to access and republish portions of it as part of a larger report – an effort savings for federated information providers who currently need to maintain the same information in multiple places. For example, professors routinely publish a list of their publications on their home page. Departments publish a list of all publications, and project pages publish a list of project-related publications from the project members. Today, someone has to manually construct these pages (presuming these federated organizations are not so tightly integrated that they maintain a shared database or other common structured information source, of course). When an author publishes a new paper or makes a correction on an existing one, he or she has to either manually update the other pages, or coordinate with the appropriate people to have all the other lists updated. WebScripter can eliminate the additional work, authors only need to mark up their personal paper publication with DAML, and the reports for the department and project-specific pages will automatically pick up the new publication (e.g. every night). WebScripter eliminates overhead not only for the organization, but also for the individual producing the information, who no longer needs to coordinate the redistribution effort. WebScripter can also enhance the flexibility and value of Web sites with large amounts of information by publishing skeleton WebScripter reports that visitors can refine to obtain customized reports. Thus, we are cautiously optimistic that WebScripter may help with the adoption of RDF(S)/DAML on the producer side as well.

[4]The notable exception are headline exchange files such as *slashdot.org/slashdot.rdf*.

6. THOUGHTS ON END-USER CONTROL OVER AUTO-ALIGNMENT

You can currently run WebScripter either in an “ignore all equivalencies” mode or in an “auto-insert all known equivalencies” mode, neither of which is ideal of course. In particular, the latter may quickly become impractical if a large number of people share alignment data, even if they are not ill-intentioned. This is either because they made a honest mistake (they aligned homepages from one ontology with email addresses from another and did not notice) or because they had a different type of equivalency in mind when they authored their report (graduate research assistants are the same as machines in the sense that they cost the project money to support, but that may then cause machines to auto-appear in a report of someone else trying to author a personnel list). We see the following potential solutions (which are not mutually exclusive).

- *Centralized Human Editors.* One possiblity is for an organization to appoint an “alignment czar”. The job of such a czar would be to periodically validate the equivalency data contributed by organization members into a staging area. If approved, equivalency files are then moved to that organization’s official equivalency data area. Cautious organization members can then exclusively make use of the approved equivalency data while adventurous ones are free to use staging data or external data. Obviously, any use of explicit human effort is associated with costs; however, one attraction of this model is that the “alignment czar” does not nearly need the technical sophistication of an “ontology librarian” and can possibly be a clerical worker given a specialized graphical application.

- *Social Filtering.* Another approach would be to keep track of the authors of equivalency statements as well as the users of equivalency statements (neither of which we currently do); this would enable users to say “I want to use the same equivalency data that Jim and Chris are using” (this is a nicely implicit way to limit equivalencies to e.g. the accounting context if they are co-workers in accounting, without having to more formally define the context, which is a more abstract and difficult task). This would also allow cautious users to express “I am willing to use any DAML equivalency file that at least 10 others are using” (which addresses the erroneous-alignment problem but not the context mismatch problem).

- *Fine-Grained Control in the User Interface.* Finally, it would be nice to have a compact display of the available equivalency information. This display would show a row of information about the available equivalency information and give the user a checkbox for incorporating or ignoring each. Table 2 sketches a preliminary design for deciding which *sameClassAs* statements to use. (This sketch assumes that we store much more fine-grained information in the equivalency files than we currently do.)

  The first column shows the human-given label of the class that is being declared as equivalent to the one the user added by hand. The second column indicates the level of indirection - 1 if the equivalency file directly states that the class the user just added by hand is the same as the class shown, 2 or more if the equivalence was inferred by transitive closure. The third column contains the Uniform Resource Locator for the equivalency file. The fourth column shows the name of the author of the WebScripter report that implied the equivalencies. The fifth column contains the number of additional rows inserted into the user’s report if she would incorporate the equivalency. The sixth column indicates when the report that resulted in the equivalency statements was authored. The last column sums up how many other users already made use of the equivalency statement in their reports.

Class	Hops	Origin	Author	Rows	Date	Users
Person	1	stanford.e...	Smith	235	10/6/02	12
Employee	1	stanford.e...	Smith	57	10/6/02	6
Staff	1	stanford.e...	Smith	697	10/6/02	0
Member	2	www.isi.e...	Chen	15	3/4/01	17
Person	2	cmu.edu/...	Miller	973	12/7/01	4
Member	2	cmu.edu/...	Miller	107	12/7/01	9

Table 2: Sketch of a graphical user interface.

7. THOUGHTS ON OTHER OPEN QUESTIONS

Addressing a number of other issues would also help in making DAML and WebScripter use take off.

- *How do ordinary users find good original Semantic Web content?* WebScripter does not address this problem: once you found one it can point you to related content that others may have by using an equivalency-aware DAML search engine such as Teknowledge’s DAML Semantic Search Service [3]. There are no Yahoo-style portals for DAML content yet to our knowledge. There are, however at least two RDF crawlers – one from BBN [2] and one from the University of Karlsruhe [5] – that could help in building such a portal.

- *What does it really mean for two classes or two attributes to be “the same”?* The current DAML equivalance statements allow users to say that x is equivalent to y. We likely need a replacement construct that allows users to express that x is equivalent to y in the sense of (or context of) z. We will try to influence the DAML language definition in that direction (but admittedly aren’t quite sure ourselves how to model z). The most difficult problem we see is in the end-user interface for stating these more complex equivalencies.

8. RELATED WORK

WebScripter’s approach to ontology alignment is extreme: terms from different ontologies are always assumed to mean different things by default, and all ontology mapping is done by humans (implicitly, by putting them into the same column of a report).

This is similar in spirit to Gio Wiederhold’s mediation approach to ontology interoperation [18], which also assumes that terms from different ontologies never mean the same thing unless committees of integration experts say they are. WebScripter pushes that concept to the brink by replacing the experts with ordinary users that may not even be aware of their implicit ontology alignment contributions. (Note, however, that we cannot yet proof that this collective alignment data is indeed a useful source for automatic ontology alignment on an Internet scale – we lack sufficient data from distributed WebScripter use to make that claim.)

The ONION system [15] takes a semi-automated approach to ontology interoperation: the system guesses likely matches between terms of two separately conceived ontologies, a human expert knowledgeable about the semantics of both ontologies then verifies the inferences, using a graphical user interface. ONION's guessing analyzes the schema information using relationships with semantics known to the system in advance (subclass-of, part-of, attribute-of, instance-of, value-of); in WebScripter human users rely purely on the *data instances* to decide what collates and what doesn't (because they are just not expert enough to analyze the abstractions). That being said, incorporating ONION-style alignment guessing into WebScripter would clearly be beneficial presuming the rate of correct guesses is sufficiently high.

OBSERVER [14], SIMS [9], TSIMMIS [11] and the Information Manifold [13] are all systems for querying multiple data sources of different schemata in a uniform way; however, they all rely on human experts to devise the ontological mappings between the sources to our knowledge. This is because they mediate between structured dynamic data sources (such as SQL/ODBC sources) without run-time human involvement where a higher level of precision is required to make the interoperation work. In contrast, WebScripter is targeted towards mediating between different ontologies in static RDF-based Web pages with run-time human involvement, where the need for precision in the translation is naturally lower.

9. EVALUATION AND CONCLUSIONS

WebScripter has turned out to be a valuable practical tool even for the simple single-ontology case where there is only one schema but the instance data is distributed over many Web pages. For example, the Distributed Scalable Systems Division at ISI automatically pulls together its people page from many different DAMLized Web pages: some information is maintained by individuals themselves (such as their research interests), other information is maintained by the division director (such as project assignments), and some information is maintained at the institute level (such as office assignments); this relieved the administrative assistant from manually maintaining everyone's interests [6]. WebScripter has also been used externally, for example to maintain a Semantic Web tools list [7]. You can download WebScripter from [8].

However, the most exciting application of WebScripter, as a world-wide collaborative ontology translation tool, is confined to experimental use by ourselves at this point. This is more due to a lack of widespread interesting RDF(S) content than it is due to any limitation of WebScripter itself. Nevertheless, we are excited about this new approach to global knowledge sharing, may it be achieved by a future version of WebScripter or a similar tool or tools. The key difference we see between "traditional" ontology translation and our approach is that non-experts perform all of the translation - but potentially on a global scale, leveraging each others' work.

10. ACKNOWLEDGMENTS

We gratefully acknowledge DARPA DAML program funding for WebScripter under contract number F30602-00-2-0576. The first author would also like to acknowledge AFOSR funding under grant number F49620-01-1-0341.

11. REFERENCES

[1] http://www.w3.org/TR/2000/CR-rdf-schema-20000327/.

[2] http://www.daml.org/crawler/.

[3] http://reliant.teknowledge.com/DAML.

[4] http://www.isi.edu/webscripter/daml-personnel.gen.html.

[5] http://ontobroker.semanticweb.org/rdfcrawl.

[6] http://www.isi.edu/divisions/div2/. Click on People.

[7] http://tools.semanticweb.org.

[8] http://www.isi.edu/webscripter.

[9] Y. Arens, C. Knoblock, and W.-M. Shen. Query reformulation for dynamic information integration. *Intelligent Information Systems*, 6(2-3):99–130, 1996.

[10] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[11] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, V. Vassalos, and J. Widom. The TSIMMIS approach to mediation: data models and languages. *Intelligent Information Systems*, 8(2):117–32, 1997.

[12] E. Hovy. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, 1998.

[13] A. Levy, D. Srivastava, and T. Kirk. Data model and query evaluation in global information systems. *Intelligent Information Systems*, 5(2):121–43, 1995.

[14] E. Mena, A. Illarramendi, V. Kashyap, and A. Sheth. OBSERVER: an approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases*, 8(2):223–71, 2000.

[15] P. Mitra and G. Wiederhold. An algebra for semantic interoperability of information sources. In *2nd Annual IEEE International Symposium on Bioinformatics and Bioengineering*, pages 174–82, Bethesda, MD, USA, November 4-6 2001.

[16] P. Mitra, G. Wiederhold, and M. Kersten. A graph-oriented model for articulation of ontology interdependencies. In *Advances in Database Technology - EDBT 2000. 7th International Conference on Extending Database Technology*, Lecture Notes in Computer Science, pages 86–100, Konstanz, Germany, March 27-31 2000.

[17] N. F. Noy and M. A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In *17th National Conference on AI*, 2000.

[18] G. Wiederhold. Interoperation, mediation, and ontologies. In *International Symposium on Fifth Generation Computer Systems, Workshop on Heterogeneous Cooperative Knowledge-Bases*, volume W3, pages 33–48. ICOT, Tokyo, Japan, December 1994.