Exploiting Semantic Web Technologies to Model Web Form Interactions *

Wolfgang Holzinger
DBAI Group
TU Wien, Austria
holzing@dbai.tuwien.ac.at

Bernhard Krüpl
DBAI Group
TU Wien, Austria
kruepl@dbai.tuwien.ac.at

Robert Baumgartner
Lixto Software GmbH
Vienna, Austria
baumgartner@lixto.com

ABSTRACT

Form mapping is the key problem that needs to be solved in order to get access to the hidden web. Currently available solutions for fully automatic mapping are not ready for commercial meta-search engines, which still have to rely on hand crafted code and are hard to maintain. We believe that a thorough formal description of the problem with semantic web technologies provides a promising perspective to develop a new class of vertical search engines that is more robust and easier to maintain than existing solutions.

In this paper, instead of trying to tackle the mapping problem, we model the *interaction* necessary to fill out a web form. First, during a user-assisted phase, the connection from the visible elements on the form to the domain concepts is established. Then, with help from background knowledge about the possible interaction steps, a plan for filling out the form is derived.

Categories and Subject Descriptors: H.3.4 [Information Storage and Retrieval]: Systems and Software

General Terms: Algorithms, Design, Experimentation.

Keywords: Hidden Web, Web Data Extraction, Web Form Mapping.

1. INTRODUCTION

In the scientific community a number of approaches towards meta-search have been proposed [3] [4]. Most of them, however, focus on deep web query probing and general information retrieval from documents accessible only via form submission.

The Metamorph project aims to generate maintainable vertical deep web search engines. Although NLP tools, query submission, result aggregation and ranking are also part of the system, we concentrate on the aspect of form interaction modelling in this paper. We favor a semi-automatic system where a human user can leverage his strengths in capturing meaning in natural texts. To support this notion, we have created a tagging tool that allows a service designer to annotate form elements with domain concepts; general HTML concepts are recognized automatically.

We will now discuss the actual form modeling that builds upon this tagging phase.

*This research is supported in part by the Austrian Forschungsförderungsgesellschaft FFG under Project Grant 812991.

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

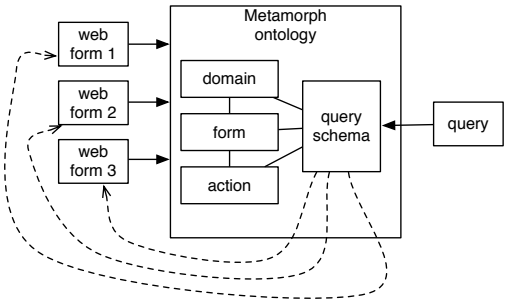

Figure 1: Query mapping via ontology

2. MODELING FORM INTERACTION

The principle of our approach is outlined in figure 1. Arbitrary web forms and meta-queries are represented in the Metamorph RDF/OWL ontology, which consists of three representational parts, a schema for queries, and a set of rules that connecting all of these. The mapping from the query to local forms can be derived by reasoning and rule application within the ontology.

Each search form on a website has its own idiosyncratic way to represent a search request. To capture the specific semantics for such a form, the Metamorph ontology must be able to faithfully represent three aspects:

- *form:* The technical details how the local search form is constructed, which HTML input elements were used in which combination and their positions on their source page for future reference.
- *domain:* A domain model rich enough to annotate each semantic reference in the local search form.
- *action:* Each HTML form element has an intrinsic action associated with it; a button can be clicked, a string value can be fed into a text field (see table 2.)

From the association of form elements with domain concepts (performed in the tagging stage), we can assign a "purpose" to the action associated with the element. If we know that an option o inside a select box refers to some specific domain individual i, the action on clicking on o has the well defined meaning, or "purpose", of submitting that specific value i to the local search request. Having these teleological facts in place, the mapping of queries to local forms is derived easily. We use RDF/OWL models to formalize the

PICK(x)	select option x in a choice field
CLICK(x)	click on element x
ENTER(s,x)	enter string s into textfield x

Table 1: User interaction primitives

Metamorph ontology, enhanced by rules as provided in the Jena toolkit.

Example: Suppose there is an select box containing month names. During tagging, the relation from the pure text strings to the "Month" concept has been established. Thus the interaction semantics for each option can be formalized:

```
pick(o1) achieves fillin(monthJanuary)
pick(o2) achieves fillin(monthFebruary)
...
pick(o6) achieves fillin(monthJune)
```

A query posed to the Metamorph system can now be interpreted as a request to fill in certain data. For example, a query for the month June can be translated into a simple SPARQL query:

```
?action achieves fillin(monthJune)
```

which will be answered by the appropriate action:

```
pick(o6)
```

But we can take this translation a step further. With the semantic information gathered in our entangled ontologies, we can actually devise a plan for mapping the query *schema*. This plan can then be further translated into any traditional programming language and executed independently of the resource-intensive reasoning framework.

Suppose a very simple global request for filling out a date consisting only of the three parameters; day, month and year. From our domain ontology we know that each parameter can have one of a finite number of values:

```
Day  == oneOf(day1, day2, ..., day31)
Month == oneOf(january, february, ..., december)
Year  == oneOf(2005, 2006, ..., 2010)
```

Because our knowledge base is saturated with the interaction rules for any value:

```
pick(d1) achieves fillin(day1)
pick(d2) achieves fillin(day2) ...
```

we can write down a plan that can deal with any instance of a global request in a straightforward manner:

```
if (Day == day1)  pick(d1)
if (Day == day2)  pick(d2)
...
if (Month == january)  pick(m1)
if (Month == february)  pick(m2)
...
if (Year == 2010) pick(y10)
```

Note that this script does not refer any more to any internal ontology details. It is purely end-to-end from the global search parameters to the interaction language. For quick evaluation, we experimented in implementing these scripts in Javascript using the Chickenfoot[2] primitives to execute the user interactions. Due to the simple nature of these plans, both translations are straightforward.

The derivation of these plans follows a simple algorithm:

```
IF C parameterOf GlobalRequest
AND x instanceOf C
AND pick(o) achieves fillin(x)
=>
assert "if (C==x) pick(o)"
```

In a similar manner, we address cases where many-to-one and one-to-many relations or value transformations are required for expressing the mapping from form elements to domain concepts.

3. IMPLEMENTATION AND VALIDATION

We verified the validity of our methodology by implementing the form modeling approach as part of an interactive and semi-automatic tagging system. Flight search forms from the air travel domain were tagged interactively and semi-automatically using the Lixto Visual Developer [1] with the domain and HTML concepts of the Metamorph ontology. We could verify that our methodology was complete enough to cover all of the 32 search forms we evaluated.

We also found that our declarative approach significantly reduced the time to add new websites to a meta-search system compared to the traditional method of manual wrapper generation. Since query plans are completely independent from the ontology at run-time, we found that the execution time needed to run queries and the robustness against layout changes was perfectly acceptable.

4. CONCLUSION

The main advantage of our declarative approach is that it modularizes the work needed to create and maintain meta-search applications into different levels: The addition of new web search forms requires just the tagging of form elements with ontology concepts and can be performed by a service designer. The skills of a trained ontology engineer are only needed to adjust the domain subontology when setting up a completely new search domain.

Our approach gives the ontology engineer and service designer a number of possibilities to efficiently create and maintain a vertical search scenario. Although we are currently incorporating more automatic mechanisms based on machine learning techniques into the tagging process, we do not envision a fully automated approach in the spirit of deep web query probing. Instead, skilled service designers will fine-tune their applications to reach a robustness and reliability that is far above that of any automatic system.

As a next step, we plan to broaden our test scenarios by expanding them to other meta-search domains such as hotel search. We will also put more emphasis on dealing with dynamical forms and online applications.

5. REFERENCES

[1] Baumgartner, R., Ceresna, M. and Ledermueller, G. (2005). Deep Web Navigation in Web Data Extraction. In *CIMCA–IAWTIC, p. 698–703* .

[2] Chickenfoot for Firefox – Rewrite the Web. *http://groups.csail.mit.edu/uid/chickenfoot/*.

[3] He, B., Zhang Z. and Chen-Chuan, K. (2005). Towards building a metaquerier: Extracting and matching web query interfaces. In *ICDE, p. 1098–1099*.

[4] Meng, W., Peng, Q. (2004). Clustering e-commerce search engines based on their search interface pages using WISE-cluster. In *WIDM, p. 231–246*.