Enhancing Wrapper Usability through Ontology Sharing and Large Scale Cooperation

Christian Schindler, Pranjal Arya, Andreas Rath, and Wolfgang Slany

Institute of Software Technology
Graz University of Technology
{cschindl,parya,arath,wsi}@ist.tugraz.at

Abstract. The htmlButler project aims at enhancing the usability of visual wrapper technology while preserving versatility. htmlButler will allow, for an untrained user who has only the most basic web knowledge, to visually specify simple but useful wrappers and, for a more tech-savvy user, to visually or otherwise specify more complex wrappers. htmlButler was started 2005/2 and is based on visual wrapping technology research carried out in the Lixto project since 2000. What is new in htmlButler is that (a) the application is entirely server based, the user accessing it through his or her standard browser, (b) because of the centralized wrapper configuration and processing, the knowledge about popular wrappers can be leveraged to facilitate the specification of wrappers for new users, and (c) users can contribute narrow and precise ontologies that help the system in recognizing potential meaning in web pages, thereby alleviating the complexity of future wrapper configurations

1 Introduction

Use cases in different vertical market domains suggest that end users (e.g., quality managers in the automotive domain, but also private end users) are eager to wrap and aggregate Web data, e.g., to notify themselves of changes in particular parts of pages of interest. However, specifying what to wrap with current interactive wrapping technology is a difficult task for untrained users. Note that the same users have little trouble communicating what they want to be wrapped to other humans, and wrappers can usually wrap that information once it has been correctly specified. The problem therefore lies in the user interface. We would like to investigate whether the communication between human users and wrapper technology can be improved through the use of high-level semantic concepts. Our aim is to make wrapper technology understand high-level concepts that human users might want to employ when wrapping web pages. Ontologies that provide systematic, computer oriented representations of real world semantic concepts and relations between them can be used to store and reason about such semantic concepts. An opportunity that arises in the realm of the Internet and in particular when we aim at widespread adoption of wrapper technology is that user communities can cooperate in building up shared ontologies. A wider audience will benefit from such an effort, thus building up a positive feedback

Vojtěch Svátek, Václav Snášel (Eds.): RAWS 2005, pp. 72–77, ISBN 80–248–0864–1.

loop with more and more users contributing to and mutually benefiting from an increasingly intelligent ontology based Web wrapper.

2 Previous Technology

htmlButler is based on the Lixto set of tools that allow application developers to implement such processes without the need for manual coding. The Lixto Visual Wrapper generation tool is based on a new method of robustly identifying and extracting relevant content parts of HTML documents and translating the content to XML format [1]. Lixto wrappers are embedded into an information processing framework, the Lixto Transformation Server[2]. The Lixto Transformation Server enables application developers to format, transform, integrate, and deliver XML data to various devices and applications. Using ontologies in wrapper specification as well as the creation of shared ontologies has been studied previously [3][4][5]. However, we are not aware of attempts to combine these two approaches. Also, while many ontology projects eventually succeed in the task of defining upper domain ontology, populating the third level, what is called the specific domain ontology, is the actual barrier that very few projects could overcome so far. We will investigate how cooperatively created, shared ontologies with corresponding new user interface paradigms could be used to overcome this barrier.

3 htmlButler

To satisfy a low inhibition threshold, we postulated the following criteria:

No Installation: The user need not install any new application or plug-in for the browser. This eliminates the concern of downloading a potentially harmful program.

Easy Configuration: It is not required to have programming or web technology knowledge such as HTML or HTTP to create a wrapper. Simple wrappers can be configured entirely in a visual interactive way. Should similar wrappers or wrapped pages exist, then the system will suggest wrapper configuration accordingly.

Usability: The user just has to enter an URL of a website, his email address and the frequency or schedule the extracted information should be sent to him. Alternatively to an exact URL, the system also will accept keywords that return a list of sites found by some search engine like google. Navigation and login protocols needed for certain non persistent URLs or password protected pages will also be available.

Resistance: The generated wrappers will be resistant to slight website changes so that users do not have to reconfigure wrappers frequently.

Easy Maintenance: Reconfiguration and enhancing of existing wrappers is done interactively with suggestions made by the system, requiring user approval.

Appropriate User Levels: A user will be able to make the most out of his skills. At the novice level, he will just be using the notification service for change. At the second stage, he would be able to enhance the simpler wrappers of the novice user. At the third and the final level, he would have the liberty to create the ontology and semantic concepts for a particular domain.

4 Semantic Supported Wrapper Generation

Narrow but detailed ontologies can enhance the process of creating wrappers in supporting the user with intelligent suggestions about extractable content of the web page the user is interested in. For instance in the system an ontology describing TV shows and typical information found therein including the names of popular artists and directors or series help a user who wants to add a new TV station in configuring the wrapper needed to map the listings to the XMLTV format, much like a human operator would recognize the various entries found on such a page. htmlButler thus emulates the human understanding of what a website is all about based on sample content. A more advanced user could semi-automatically add new or missing concepts to the ontology based on his or her current needs to make the wrapper generation easier, from which other users might profit in the future. Already working wrappers can be reused and enhanced and work as a base for more complex usage scenarios. The ontology can be used for verifying the results of the created wrapper and adjust the wrapper to changes in the web site if for instance a column is inserted to a TV listing table and the program description has thereby changed place.

5 Shared Ontologies

So called extraction ontologies can help to make the wrapper configuration process easier[6]. The creation of ontologies is usually hard work, but not for an expert. In our project we aim at letting non expert users combine and manipulate public contributions to centrally maintained shared ontologies, and giving experts a easy to use graphical tool to create complex extraction tools. Even more, incompletely contributing narrow domain ontologies will still make sense, since a specific goal that is of interest to the contributing user, will be served. A related problem that immediately arises with shared ontologies is quality. However, we will experimentally investigate whether sacrificing quality for usefulness will be sufficient for efficient extraction purposes. To somehow recover practical quality, we will investigate self correction systems where users, e.g., can rate the usefulness of ontologies contributed by others. These ratings can then be used to influence the way the wrapper will suggest tokens that should be wrapped on similar occasions. We will therefore study existing tools for their suitability for non expert, casual use ontology contribution, editing, and assessment. One thing we will need to address is supporting the group of actively contributing users and of passively consuming users. We will study whether this can be achieved by groupware tools that can be freely self organized such as WikiWiki-Webs that

are considered efficient knowledge management and knowledge sharing tools. This would greatly facilitate free format organizational communication, allowing volunteer participation in organizing FAQ lists, documentation, tutorials, how-to guidelines, feature brainstorming boards, ontology annotation repositories, and discussion places for contributed ontologies, all in a persistent manner that encourages neutral point of view consensus finding as, e.g., in the Wikipedia project (http://wikipedia.org/).

6 htmlButler Status Quo

Technically, htmlButler is almost completely server based only selecting the area of interest on the web page, which should be wrapped, is done on the client side using JavaScript. At the htmlButler start page, the user will be prompted for login or create a new user with the information, where to send the wrapping output and the default frequency of wrapper invocation. The user enters the webpages address (see Fig.1). This URL is submitted to the server, the content of this URL is fetched and processed in a certain way and sent back to the users browser embedded into a frameset (see Fig.2). The user can then select the area of interest and submit this selection to the server. At the server side the selection is processed and, by means of implicitly stored rules and semantics about the structure of web pages, a wrapper is generated and scheduled for invocation. At the actual implementation the user is not able to alter or enhance these rules. As the htmlButler project reaches its next stage, it would be possible for people to log in as a super user (or an enhanced user with more liberties to create complex wrappers). It will be a challenge to improve the user interface in that way to enable the user enhancing or submitting new concepts for the wrapper generation. htmlButler combines two basic mechanisms for data extraction tree and string extraction. For tree extraction the elements are identified with their corresponding tree path and possibly with some properties of the elements themselves. For string extraction the leaves of the HTML parse tree are examined in detail for changes.

7 Conclusion

We are interested in studying the preconditions necessary to allow the use of knowledge representation formats in wrapper specification user interfaces. We also study means to allow users to easily share their knowledge through existing knowledge representation formats used in other semantic Web applications in order to leverage their contribution efforts so that economy of scale effects can take place. A preliminary problem that is however fundamental to the success of our research agenda consists in the necessity to make wrapper user interfaces accessible to a wider audience, triggering economy of scale effects in the long run. We will therefore research usability aspects for a widespread acceptance of wrapper technology. We believe it is necessary to investigate the particular aspects of ontology sharing and ontology usage in wrapper applications. We will

Fig. 1. Submission of the URL which should be wrapped

Fig. 2. Submission of the selected area

in particular investigate whether the interaction between untrained users and wrapper generators can be enhanced by community-shared, -maintained, and -validated ontologies. The outcome of this research should be a set of new methods, algorithms, and heuristics that allow the system to efficiently use shared ontologies in an interactive wrapper specification process, resulting in enhanced usability. Additional research will concern guidelines for both the ontology based wrapping process as well as the management of shared ontologies.

Acknowledgements

The htmlButler project is funded by the Austrian FFG under the FIT-IT program line as a part of the NextWrap project.

References

1. R. Baumgartner, S. Flesca, G.G.: Visual web information extraction with lixto. In: Proceedings of VLDB. (2001)
2. M. Herzog, G.G.: Infopipes: A flexible framework for m-commerce applications. In: Proceedings of TES workshop at VLDB, 2001. (2001)
3. M. Hatala, G.: Global vs. community metadata standards: Empowering users for knowledge exchange. In: Proceedings of the First International Semantic Web Conference. (2002)
4. M. Missikoff, X.W.: Consys - a group decision-making support system for collaborative ontology building. In: Proceedings of Group Decision & Ne-gotiation 2001 Conference. (2001)
5. M. Stonebraker, J.H.: Content integration for e-business. In: ACM Sigmod Conference. (2001)
6. D.W. Embley, C.T., Liddle, S.: Automatically extracting ontologi-cally specified data from html tables with unknown structure. In: Proceedings of the 21st International Conference on Conceptual Modelling. (2002)