LixtoForAll: Consumer-level Semantic Web-Squeezing and Aggregation

Wolfgang Slany
Technische Universität Graz, Austria
slany@ist.tu-graz.ac.at
Lixto Software GmbH, Austria
slany@lixto.com

Abstract

Using legacy data and services on the web requires intelligent interpretation and thus cannot be easily delegated to programs yet. Nevertheless, one would often benefit from having an electronic helper which replaces oneself in web interactions that need to be done repeatedly. To be ready for consumer-level usage, a helper such as LixtoForAll must be as easy to use and as intelligent as possible. Ontologies help LixtoForAll to better understand the user as well as the data found on the web. LixtoForAll is based on advanced visual data extraction and aggregation and will soon be available.

1 Usage scenarios

Imagine a teenager who wants to be notified when a certain popular star appears on TV. Using LixtoForAll, the teenager can interactively create a new service that does this. The teenager does not write any code as this push service can be realized in a purely visual way, with immediate feedback for the user what it will do.

Imagine a sports fan who wants a personal web page small enough for a mobile phone screen that lists all recent and future soccer games of certain teams he is interested in. Using LixtoForAll, it is straightforward to create this pull page that will be brought up to date whenever he accesses it.

Imagine a business woman (or a researcher) and frequent flyer who needs to be notified via sms about any delay, any change in the expected departure time or other news concerning her flight: without any coding, she sets up a service that will grab corresponding data for her from airport websites. She will then be able to enter her flight number through a simple web-interface. LixtoForAll will scan the sites for relevant information about the particular flight in frequent intervals and notify her in case of newsworthy changes.

Other users profit from the pioneers that first create LixtoForAll applications as these become available. All users of LixtoForAll profit from ontological data donated by other users on a need-driven basis to the system. All donated data is peer rated and positive contributions are rewarded based on their ratings.

2 Lixto

The information and functionality on the Web is a valuable source for all kinds of application domains, including reliable competition monitoring, added-value content aggregation, enriching enterprise data with content from Web resources, real-time Web data querying, automatically navigating the Deep Web, portal content integration, and mobile applications.

LixtoForAll is based on the *Lixto* set of tools that allows application developers to implement such processes without the need for manually coding. The Lixto platform provides tools to access, transform, and syndicate exactly the information needed as structured. Wrapper technology is used to extract the relevant information from HTML documents and translate it into XML which later on can be automatically processed. 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 [Baumgartner *et al.*, 2001]. Predefined syntactic (dates, currencies etc) and semantic (wordnet and geographical ontological databases) concepts allow high-level specification of what data should be extracted. Lixto is particularly well-suited for visual and interactive creation of HTML to XML wrappers.

Lixto wrappers are embedded into an information processing framework, the *Lixto Transformation Server* [Herzog and Gottlob, 2001]. The Lixto Transformation Server enables application developers to format, transform, integrate, and deliver XML data to various devices and applications. The Transformation Server features a set of predefined software components that can be used to implement XML data flow applications. Each of those components features input and output channels – along with a transport protocol – to pass XML documents to subsequent components. Within the Lixto Transformation Server the application development process relies fully on the visual paradigm. XML data flow processes can be modelled using a graphical user interface. The process structures are mapped on specific components which execute the required XML transformation operations during run-time. The components exploit XSLT as the mechanism to perform the XML document transformation operations. Required

XSLT stylesheets are generated automatically based on the configuration of the components.

Lixto Visual Wrapper can interactively and visually identify relevant chunks of data on sample pages. Non-technical users can easily create wrappers with Lixto. Internally, the system creates an extraction program to be used on similarly structured pages on which the content changes over time. The internal language is a declarative logic-based and very expressive language that also supports recursive queries useful for traversing chained result lists (e.g., links to "next 10 items" pages) [Gottlob and Koch, 2002], and optionally wrappers can directly be specified using this language. However, the visual wrapper generation covers all aspects of the language and no user is ever forced to switch to this mode. The wrapper operator can open a sample page in the internal browser window of Lixto. In some cases, it is necessary to visually create a so-called navigation sequence, storing cookies and lists of requests for pages which cannot be accessed directly.

The wrappers are embedded into the Lixto Transformation Server. This Transformation Server provides a framework for creating an information flow. The Transformation Server allows application designers to visually define and test applications using the XML processing capabilities of the deployed components. The components are as follows: The *Lixto Source* navigates to the desired Web pages used for wrapping, dealing with get and post sequences, cookies, Java Script etc., and extracts data from the web pages into XML using the previously defined wrappers. The *Lixto Integrator* aggregates heterogeneous output from different wrapper programs. The *Lixto Transformer* restructures, queries and joins XML documents; moreover, clients can subscribe to services using different queries for personalization. The *Lixto Deliverer* syndicates information, that is, formats the XML data for different client devices and protocols including cHTML, email, or sms, using either push or pull techniques. Each component can independently be configured, and has a particular input/output behavior. Components are connected with arrows – each arrow resembles an XML information flow.

A sample Lixto application is a mobile radio station that extracts current charts and play lists of radio stations for mobile devices (UMTS phones and PDAs). Another application retrieves company stock information, a third one is a personal SMS agent for airline passengers [Herzog and Gottlob, 2001], and others include personalized news agents. With Lixto's UMTS scenario "Nowplaying", which was developed in cooperation with T-Mobile, the playlist of various radio stations is extracted from Web sites in real time and integrated into a mobile portal. Furthermore, it is possible to listen to the live audiostream of the currently aired songs on mobile devices, view information on the title, the artist, and, if available on the Web, read the lyrics. Images of the CD covers are offered along with the current ranking of the selected song in various national and international charts. This application has been stable for over one year, and only two wrappers had to be re-adjusted during this time, although several changes in the Web pages occurred, due to the content-oriented wrapping style enhanced by the integrated syntactic and semantic concepts supported by Lixto Visual Wrapper, showcasing the robustness of the wrappers which compares very positively to other approaches. This and many more use-cases justify the extraction and integration technology offered by Lixto in a world where unstructured Web pages still offer most of the information.

3 LixtoForAll

In LixtoForAll, the Lixto Visual Wrapper and Lixto Transformation Server are simplified to support the most common use-cases and unclutter the access to the system for the uninitiated end user. More complex situations that can be handled with the full version of Lixto may be supported in the future if needed. The intent behind making LixtoForAll available for personal use is to get visibility for the full version of the Lixto tools, so as much functionality as possible will be made available, under the constraint of making the system as easy to use as possible. In expert mode it will always be possible to create arbitrarily complex wrappers and information flows manually.

The simplified visual wrapper and the simplified transformation server run on a publicly accessible server where users can create wrappers and information flows under an anonymous account. Certain variables remain private, but the overall scripts (wrappers and flows) become public under the GPLL. There is a reward system for popular wrappers. A wiki community web enriched with some automatically generated pages provides space to organize the functionality of LixtoForAll, e.g. in a FAQ, in tutorials, in discussion pages of publicly available wrappers and information flows, in feature brainstormings, etc.

The semantic database is so far populated only from wordnet and some public ontological databases. Here again, popular contributions will be rewarded. Standard ontological interfaces will be supported in the future.

References

[Baumgartner *et al*., 2001] R. Baumgartner, S. Flesca, and G. Gottlob. *Visual Web Information Extraction with Lixto*. In Proc. of VLDB, 2001.

[Gottlob and Koch, 2002] G. Gottlob and C. Koch. *Monadic Datalog and the Expressive Power of Languages for Web Information Extraction*. In Proc. of PODS, 2002.

[Herzog and Gottlob, 2001] M. Herzog and G. Gottlob. *InfoPipes: A flexible framework for M-Commerce applications*. In Proc. of TES workshop at VLDB, 2001.

[Baumgartner *et al*., 2004] R. Baumgartner, G. Gottlob, M. Herzog and W. Slany. *Interactively Adding Web Service Interfaces to Existing Web Applications*. In Proc. of SAINT, 2004.