An Automatic Data Grabber for Large Web Sites

Valter Crescenzi [1], Giansalvatore Mecca [2],
Paolo Merialdo [1], Paolo Missier [1]

[1] Università Roma Tre
Roma - Italy
[crescenz,merialdo]@dia.uniroma3.it,
pmissier@acm.org

[2] Università della Basilicata
Potenza - Italy
mecca@unibas.it

Abstract

We demonstrate a system to automatically grab data from data intensive web sites. The system first infers a model that describes at the intensional level the web site as a collection of classes; each class represents a set of structurally homogeneous pages, and it is associated with a small set of representative pages. Based on the model a library of wrappers, one per class, is then inferred, with the help an external wrapper generator. The model, together with the library of wrappers, can thus be used to navigate the site and extract the data.

1 Introduction

This paper presents a system to automatically grab data from the web. The target of our system is that relevant portion of the web consisting of large data intensive web sites. Prominent examples are popular e-commerce web sites, financial web sites, the web sites of relevant scientific institutions, or the web site of events of worldwide interest. These sites represent rich and up-to-date information sources. However, since they mainly deliver data through an intricate hypertext collection of HTML documents, it is not easy to build applications that access to and compute over these sources.

A partial solution to the problem comes from some recent research proposals. Based on the observation that many web sites contain large collections of structurally similar pages, several researchers have developed techniques to automatically infer web wrappers [1, 3, 4, 9], i.e., programs that extract data from HTML pages, and transform them into a machine processable format, typically in XML. These techniques take as input a small set of sample pages exhibiting a common template, and generate as output a wrapper that can be used to extract the data from any page sharing the same structure as the input samples.

These proposals represent an important step towards the automatic extraction of data from web data sources. However, as argued in [1, 4], intriguing issues arise when scaling up from the single collection of pages to whole sites. The main problems, which significantly affect the scalability of the wrapper approach, are how to identify the structured regions of the target site, and how to collect the sample pages to feed the wrapper generation process. Presently, these tasks are done manually.

The subject of our demonstration is a system that addresses these issues, making it feasible to automatically extract data from large data intensive web sites. Our system explores a target web site and automatically infers a model for the site. To reduce the number of visited pages, site exploration is driven by an adaptive approach. The inferred model describes the web site as a directed graph: nodes are classes of structurally homogeneous pages, while arcs represent collections of links among pages that belong to different classes. Each class is associated with a small set of representative pages, that can be used to infer a library of wrappers with the help an external wrapper generator.[1] The model, together with the associated wrappers, can be used to continuously extract data from the target web sites.

2 Related work

To the best of our knowledge the only system that is similar to ours in spirits is that proposed by Kao et

[1] We use ROADRUNNER, our wrapper generator module [4, 5].

Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004

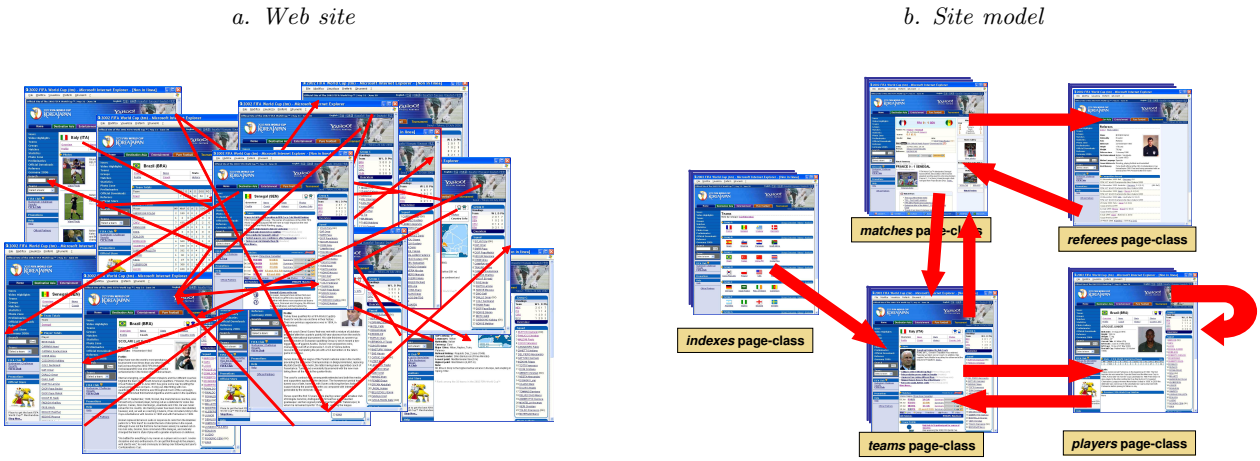

Figure 1: A site model example

al. [6]. They have have developed an entropy based technique for analyzing news web sites. Namely the goal of their proposal is to identify, within a news web site, pages of indexes to news, and pages containing news. Compared to our approach, Kao et al. only distinguish two predefined classes of pages (indexes and content), in a specific domain (news web site). On the contrary we aim at classifying pages according to their structure, without any a priori assumption.

Another field that is somehow related to our research is that of focused-crawling [2]. The goal of a focused crawler is to selectively seek out pages that are relevant to a pre-defined set of topics possibly from many different web sites. Our system can be considered as a crawler focused on recognizing structure (for extraction purposes) rather than looking for topic-relevant pages. However, we conjecture that the intensional description of a web site that we can infer could be profitably combined with approaches that crawl sites by topics.

Finally, it is worth mentioning that our approach does not crawl data behind forms. In order to extract data also from the hidden web, our system could cooperate with specific techniques, such as those proposed in [8, 7].

3 An overview of the system approach

The goal of the system is to infer a model, that describes at the intensional level the overall structure of the site in terms of classes of pages and navigational paths among them. We expect that structurally similar pages are grouped into classes, and that the link between pages can be grouped into classes of links between classes. Also, we aim at generating the model efficiently, i.e. by visiting a small portion of the target site: while building the model, the system adaptively chooses pages to visit in order to infer a complete model for the site.

Figure 2: A player page from `fifaworldcup.com`

Figure 1 shows an example of the approach: given a large web site composed by thousands of interconnected page, the system produces a model, that describes at the intensional level the structure of the site.

We illustrate the intuition behind our approach by means of an example. Consider the official FIFA 2002 world cup web site,[2] whose roughly 20,000 pages contain information about teams, players, matches, and news. Both the contents and the topology of the site are structured in a regular way: there is one page for each player, one page for each team, and so on; the links between pages reflect the semantic relationships between contents: for instance, every team page con-

[2] `http://fifaworldcup.yahoo.com/02`

tains links to the pages of its players.

A key observation in our approach is that in data-intensive web sites links reflect both the internal structure of pages and the topological structure of the site. Whenever two (or more) pages contain links that share the same layout and presentation properties, then it is likely that the two pages share the same structure. This can be more easily observed by grouping the links into (possibly singleton) *collections*: each collection is characterized by uniform layout and presentation properties.[3] Consider for example Figure 2, which shows the web page of a player from Fifaworldcup.com: every player page (they are all generated from the same template) contains the same link collections shown in this page (e.g. one link collection on the right leading to the other player of the same team, and one link collection on the top pointing to other pages about its team, and so on).

In addition, we observe that usually links within the same collection are organized either as a list of link pointing to similar pages, or as a tuple of links leading to pages with different structures. Again from our example, we have that every player page contains one link collection on the right, which is a list of links to other player pages, one link collection on the top pointing to several different kinds of page about a team ("overview", "profile", "stats" ...), etc..

Our approach for inferring a model describing the structure of a web site builds on the above observations. We model pages as objects containing tuples and list of links; a *page class* models sets of pages; links are typed: the type corresponds to the class of the target page. To infer a model, the system considers pages from the target site, and groups them into classes considering several alternative partitions, which are then ranked according to a metrics of quality.

The inference process is performed incrementally. The system starts from a given entry point (e.g. the home page), which becomes the first member of the first class in the model; then, it refines the model by exploring its boundaries to gather new pages. At each iteration, the system selects a link collection from the model outbound and iteratively fetches a page by following one of the link in the collection. In order to reduce the number of pages actually visited, after each download the system makes a guess on the class of remaining pages. If looking at the pages already downloaded there is sufficient evidence that the guess is right, the remaining pages of the collections are assigned to classes without actually fetching them. The process iterates until all the link collections are typed with a known class.

[3]The layout and presentation properties of each link collection can be described in terms of their paths on the DOM tree.

4 The Demonstration

During the demonstration we will show the system at work on some real-life web sites. Figure 3 depicts the main components of our system and their interaction.

Our demonstration will focus on the main steps of our approach for grabbing data from large web sites. Taking as input an entry point (e.g. the home page) of a target site, we will demonstrate:

Automatic web site model generation: we will show how the system is able to build a model of the site even visiting a small number of pages. In particular, we will demonstrate how the system adaptively chooses pages to visit in order to create a complete intensional description of the hypertext structure; also, we will demonstrate that large web sites containing thousands of pages can be indeed modelled with a small number of classes;

Wrapper generation: the model generated in the previous step will be used for feeding the automatic wrapper generator. The system builds a library of wrappers for the target web site, taking as input the pages from each class of the model. To perform this step, we use our wrapper generator system [5, 4]; however, other wrapper generators, such as those described in [1, 3, 9], may be used.

Data extraction: the site model, together with the wrappers, will then be used to extract data from the site. We will show that, with the help of the model and of the associated wrappers, the system provides a structured virtual view of the site.

Selective data grabbing: based on the model of the site, the user can select page classes containing data of interest. The system navigates the actual site to reach the target pages, then it applies the associated wrappers and extract the requested data.

References

[1] Arasu, A., and Garcia-Molina, H. Extracting structured data from web pages. In *ACM SIGMOD International Conf. on Management of Data (SIGMOD 2003), San Diego, California* (2003).

[2] Chakrabarti, S., van den Berg, M., and Dom, B. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks (Amsterdam, Netherlands) 31*, 11–16 (1999), 1623–1640.

[3] Chang, C.-H., and Shao-Chen, L. Iepad: Information extraction based on pattern discovery. In *Proceedings of the tenth international conference on World Wide Web* (2001).

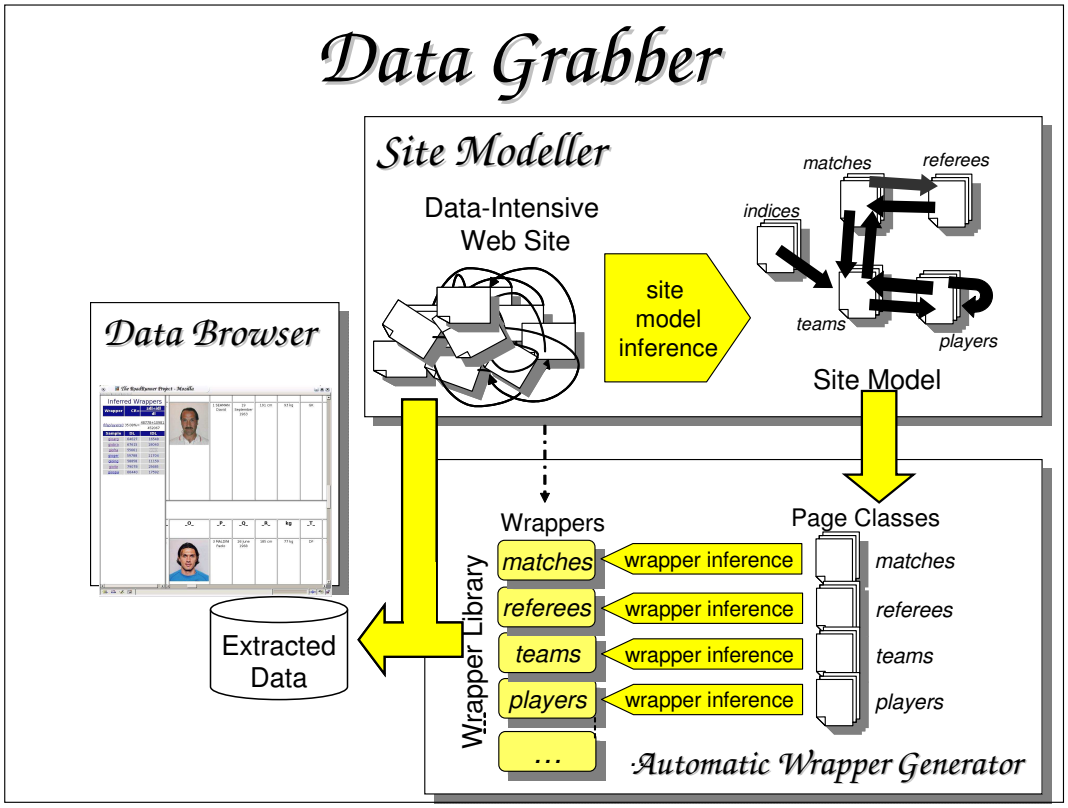

Figure 3: System components and their interaction

[4] Crescenzi, V., Mecca, G., and Merialdo, P. roadRunner: Towards automatic data extraction from large Web sites. In *International Conf. on Very Large Data Bases (VLDB 2001), Roma, Italy, September 11-14* (2001).

[5] Crescenzi, V., Mecca, G., and Merialdo, P. Roadrunner: Automatic data extraction from data-intensive web sites. In *ACM SIGMOD International Conf. on Management of Data (SIGMOD'2002), Madison, Wisconsin* (2002).

[6] Kao, H., Lin, S., Ho, J., and M.-S., C. Mining web informative structures and contents based on entropy analysis. *IEEE Transactions on Knowledge and Data Engineering 16*, 1 (January 2004), 41–44.

[7] Palmieri, J., da Silva, A., Golgher, P., and Laender, A. Collecting hidden web pages for data extraction. In *ACM WIDM 2002* (2002).

[8] Raghavan, S., and Garcia-Molina, H. Crawling the hidden web. In *International Conf. on Very Large Data Bases (VLDB 2001), Roma, Italy, September 11-14* (2001).

[9] Wang, J., and Lochovsky, F. Data-rich section extraction from html pages. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE 2002), 12-14 December 2002, Singapore, Proceedings* (2002), IEEE Computer Society, pp. 313–322.