DIADEM: Domain-centric, Intelligent, Automated Data Extraction Methodology*

Tim Furche, Georg Gottlob, Giovanni Grasso, Ömer Gunes, Xiaonan Guo,
Andrey Kravchenko, Giorgio Orsi, Christian Schallhart, Andrew Sellers, Cheng Wang

Department of Computer Science, Oxford University, Wolfson Building, Parks Road, Oxford OX1 3QD
firstname.lastname@cs.ox.ac.uk

ABSTRACT

Search engines are the sinews of the web. These sinews have become strained, however: Where the web's function once was a mix of library and yellow pages, it has become the central marketplace for information of almost any kind. We search more and more for objects with specific characteristics, a car with a certain milage, an affordable apartment close to a good school, or the latest accessory for our phones. Search engines all too often fail to provide reasonable answers, making us sift through dozens of websites with thousands of offers—never to be sure a better offer isn't just around the corner. What search engines are missing is understanding of the *objects* and their *attributes* published on websites.

Automatically identifying and extracting these objects is akin to alchemy: transforming unstructured web information into highly structured data with near perfect accuracy. With DIADEM we present a formula for this transformation, but at a price: DIADEM identifies and extracts data from a website with high accuracy. The price is that for this task we need to provide DIADEM with extensive knowledge about the ontology and phenomenology of the domain, i.e., about entities (and relations) and about the representation of these entities in the textual, structural, and visual language of a website of this domain. In this demonstration, we demonstrate with a first prototype of DIADEM that, in contrast to alchemists, DIADEM has developed a viable formula.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: On-line Information Services—*Web-based services*

General Terms

Languages, Experimentation

Keywords

data extraction, deep web, knowledge

*The research leading to these results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement DIADEM, no. 246858, `http://diadem-project.info/`.

WWW'12 Apr 16-20, 2012 Lyon, France.

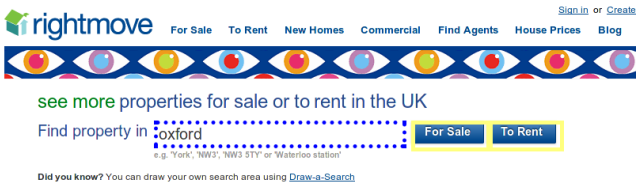


Figure 1: Exploration: Form

1. INTRODUCTION

Search is failing. In our society, search engines play a crucial role as information brokers. They allow us to find web pages and by extension businesses, articles, or information about products wherever they are published. Visibility on search engines has become mission critical for most companies and persons. However, for searches where the answer is not just a single web page, search engines are failing: What is the best apartment for my needs? Who offers the best price for this product in my area? Such *object queries* have become an exercise in frustration: we users need to manually sift through, compare, and rank the offers from dozens of websites returned by a search engine. At the same time, businesses have to turn to stopgap measures, such as aggregators, that provide customers with search facilities in a specific domain (*vertical search*). This endangers the just emerging universal information market, "the great equalizer" of the Internet economy: visibility depends on deals with aggregators, the barrier to entry is raised, and the market fragments. Rather than further stopgap measures, the "publish on your website" model that has made the web such a success and so resilient against control by a few (government agents or companies) must be extended to object search: Without additional technological burdens to publishers, users should be able to search and query objects such as properties or laptops based on attributes such as price, location, or brand. Increasing the burden on publishers is not an option, as it further widens the digital divide between those that can afford the necessary expertise and those that can not.

DIADEM, an ERC advanced investigator grant, is developing a system that aims to provide this bridge: Without human supervision it finds, navigates, and analyses websites of a specific domain and extracts all contained objects using highly efficient, scalable, automatically generated wrappers. The analysis is parameterized with domain knowledge that DIADEM uses to replace human annotators in traditional wrapper induction systems and to refine and verify the generated wrappers. This domain knowledge describes the ontology as well as the phenomenology of the domain: what are the entities and their relations as well as how do they occur on websites. The latter describes that, e.g., real estate properties include a location and a price and that these are displayed prominently.

Figures 1 and 2 show examples (screenshots from the current prototype) of the kind of analysis required by DIADEM for fully

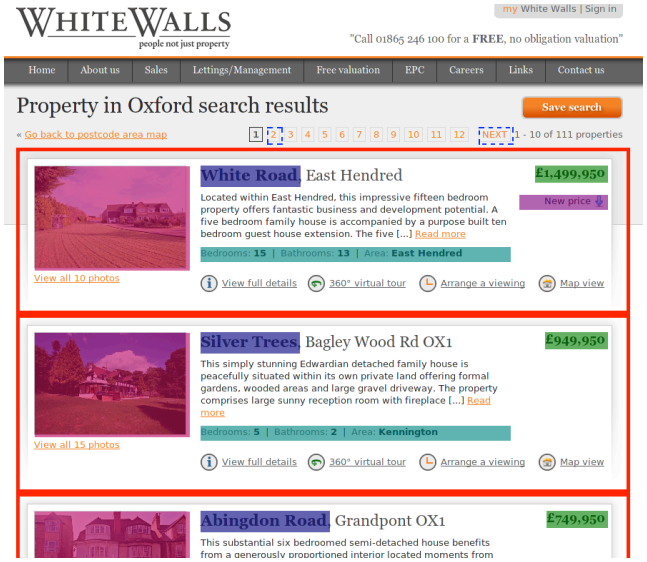


Figure 2: Identification: Result Page

automated data extraction: Web sites needs to be explored to locate relevant data, here real estate properties. This includes in particular forms as in Figure 1 which DIADEM automatically classifies such that each form field is associated with a type from the domain ontology such as "minimum price" field in a "price range" segment. These form models are then used for filling the form for further analysis, but also to generate exhaustive queries for latter extraction of all relevant data. Figure 2 illustrates a (partial) result of the analysis performed on a result page, i.e., a page containing relevant objects of the domain: DIADEM identifies the objects, their boundaries on the page, as well as their attributes. Objects and attributes are typed according to the domain ontology and verified against the domain constraints. In Figure 2, e.g., we identify real estate properties (for sale) together with price, location, legal status, number of bed and bathrooms, pictures, etc. The identified objects are generalised into a wrapper that can extract such objects from any page following the same template without further analysis.

In the first year and a half, DIADEM has progressed beyond our expectations: The current prototype is already able to generate wrappers for most UK real estate and used car websites with higher accuracy than existing wrapper induction systems. In this demonstration, we outline the DIADEM approach, first results, and describe the live demonstration of the current prototype.

2. DIADEM—THE METHODOLOGY

DIADEM is fundamentally different from previous approaches. The integration of state-of-the-art technology with reasoning using high-level expert knowledge at the scale envisaged by this project has not yet been attempted and has a chance to become the cornerstone of next generation web data extraction technology.

Standard wrapping approaches [16, 6] are limited by their dependence on templates for web pages or sites and their lack of understanding of the extracted information. They cannot be generalised and their dependence on human interaction prevents scaling to the size of the Web. Several approaches that move towards fully automatic or generic wrapping of the existing World Wide Web have been proposed and are currently under active development. Those approaches often try to iteratively learn new instances from known patterns and new patterns from known instances [5, 18] or to find generalised patterns on web pages such as record boundaries. Although current web harvesting and automated information extraction systems show significant progress, they still lack the combined recall and precision necessary to allow for very robust queries. We intend to build upon those approaches, but add deep knowledge into the extraction process and combine them in a new manner into our knowledge-based approach; we propose to use them as low-level fact finders, whose output facts may be revised in case more accurate information arises from other sources.

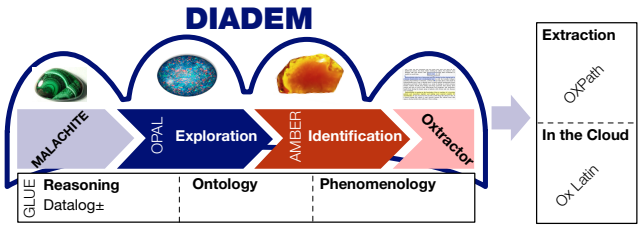


Figure 3: DIADEM Overview

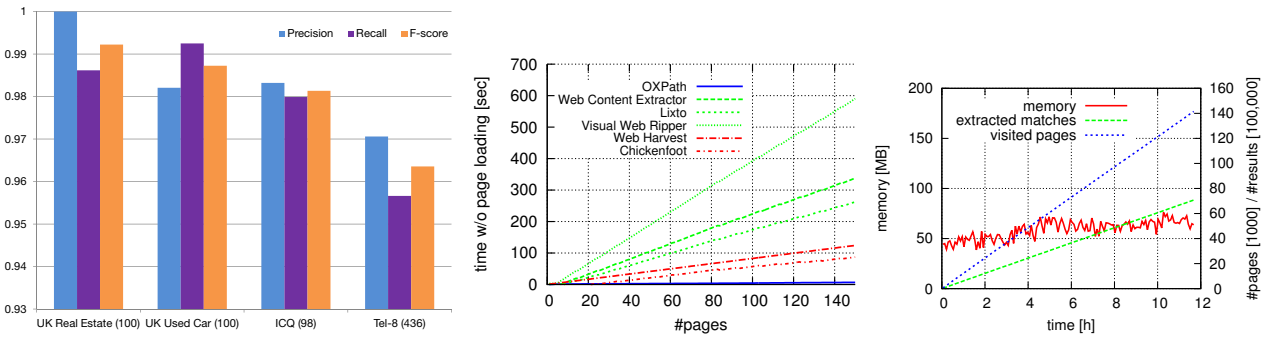
DIADEM focuses on a dual approach: at the low level of web pattern recognition, we use machine-learning complemented by linguistic analysis and basic ontological annotation. Several of these approaches, in addition to newly developed methods, are implemented as lower level building blocks (fact finders) to extract as much knowledge as possible and integrate it into a common knowledge base. At a higher level, we use goal-directed domain-specific rules for reasoning on top of all generated lower-level knowledge, e.g. in order to identify the main input mask and the main extraction data structure, and to finalize the navigation process.

DIADEM approaches data extraction as a two stage process: In a first stage, we analyse a small fraction of a web site to generate a wrapper that is then executed in the second stage to extract all the relevant data on the site at high speed and low cost. Figure 3 gives an overview of the high-level architecture of DIADEM. On the left, we show the analysis, on the right the execution stage.

(1) Sampling analysis: In the first stage, a sample of the web pages of a site are used to *fully automatically* generate wrappers (i.e., extraction program). The analysis is based on domain knowledge that describes the objects of interest (the "ontology") and how they appear on the web (the "phenomenology"). The result of the analysis is a wrapper program, i.e., a specification how to extract all the data from the website without further analysis.
Conceptually, it is divided into two major phases, though these are closely interwoven in the actual analysis:

(i) *Exploration:* DIADEM automatically explores a site to locate relevant objects. The major challenge here are web forms: DIADEM needs to understand such forms enough to fill them for sampling, but also to generate exhaustive queries for the extraction stage such that *all* the relevant data is extracted (see [1]). DIADEM's form understanding engine OPAL [9] is uses an phenomenology of forms in the domain to classify the form fields.
The exploration phase is supported by the page and block classification in MALACHITE where we identify, e.g., next links in paginate results, navigation menus, and irrelevant data such as advertisements. We further cluster pages by structural and visual similarity to guide the exploration strategy and to avoid analysing many similar pages. Since such pages follow a common template, the analysis of one or two pages from a cluster usually suffices to generate a high confidence wrapper.

(ii) *Identification:* The exploration unearths those web pages that contain actual objects. But DIADEM still needs to identify the precise boundaries of these objects as well as their attributes. To that end, DIADEM's result page analysis AMBER [10] analyses the repeated structure within and among pages. It



(a) OPAL (b) OXPath Comparative (c) OXPath Scaling

Figure 4: Exploration and Execution

exploits the domain knowledge to distinguish noise from relevant data and is thus far more robust than existing data extraction approaches. AMBER is complemented by Oxtractor, that analysis the free text descriptions. It benefits in this task from the contextual knowledge in form of attributes already identified from AMBER and of background knowledge from the ontology.

(2) Large-scale extraction: The wrapper generated by the analysis stage can be executed independently and repeatedly. We have developed a new wrapper language, called OXPath [12], the first wrapper language for large scale, repeated (or even continuous) data extraction. OXPath is powerful enough to express nearly any extraction task, yet as a careful extension of XPath maintains the low data and combined complexity. In fact, it is so efficient, that page retrieval and rendering time by far dominate the execution. For large scale execution, the aim is thus to minimize page rendering and retrieval by storing pages that are possibly needed for further processing. At the same time, memory should be independent from the number of pages visited, as otherwise large-scale or continuous extraction tasks become impossible. With OXPath we manage to obtain all these characteristics, as shown in Section 3. For a more detailed description of DIADEM's stages, see [11].

DIADEM's analysis uses a knowledge driven approach based on a domain ontology and phenomenology. To that end, most of the analysis is implemented in logical rules on top of a thin layer of fact finders. For the reasoning in DIADEM we are currently developing a reasoning language targeted at highly dynamic, modular, expressive reasoning on top of a live browser. This language, called GLUE, builds on Datalog$^{\pm}$ [4, 15, 2, 3, 17] , DIADEM's ontological query language. Datalog$^{\pm}$ allows us to reason on top of fairly complex ontologies including value invention and equality dependencies with little performance loss over basic datalog (see [7] for a current state of datalog engines). We are also working on probabilistic extensions [13, 14] of Datalog$^{\pm}$ for use in DIADEM.

3. FIRST RESULTS

To give an impression of the achievements of DIADEM in its first year, we briefly summarise results on three components of DIADEM: its form understanding system, OPAL; its result page analysis, AMBER; and the OXPath extraction language.

Figures 4a and 5 report on the quality of form understanding and result page analysis in DIADEM's first prototype. Figure 4 [9] shows that OPAL is able to identify about 99% of all form fields in the UK real estate and used car domain correctly. We also show the results on the ICQ and Tel-8 form benchmarks, where OPAL achieves $>$ 96% accuracy (in contrast recent approaches achieve at best 92% [8]). The latter result is without use of domain knowledge. With domain knowledge we could easily achieve close to 99% accuracy also in these cases. Figure 5 [10] shows the results for data area, record, and attribute identification on result pages for AMBER in the UK real estate domain. We report each attribute separately. AMBER achieves on average 98% accuracy for all these tasks, with a tendency to perform worse on attributes that occur less frequently (such as the number of reception rooms).

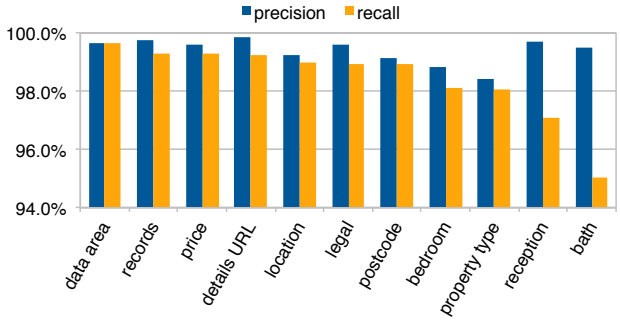


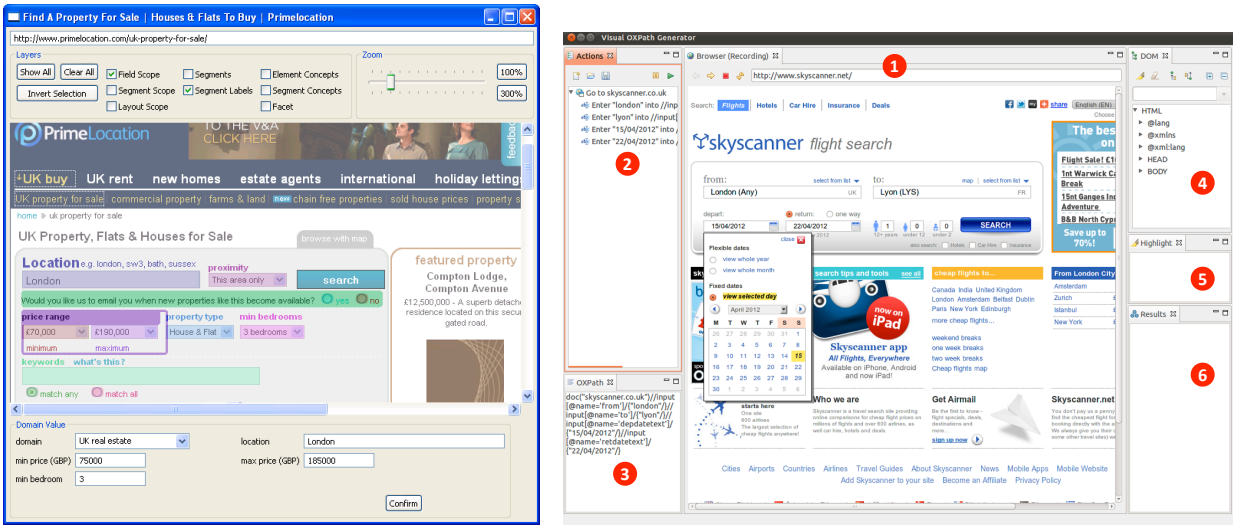
Figure 5: Identification: in UK real estate

Figures 4b and 4c summarise the results for OXPath: It easily outperforms existing data extraction systems, often by a wide margin. Its high performance execution leaves page retrieval and rendering to dominate execution ($>$ 85%) and thus makes avoiding page rendering imperative. We minimize page rendering by buffering any page that may still be needed in further processing, yet manage to keep memory consumption constant in nearly all cases, as evidenced by Figure 4c where we show OXPaths memory use over a 12h extraction task extracting millions of records from hundreds of thousands of pages. For more details see [12].

4. DIADEM DEMONSTRATION

As its approach, the demonstration of DIADEM is split into two parts. The first part illustrates its analysis using the current DIADEM prototype. The second part illustrates wrapper execution with OXPath using a combination of web demos and visual wrapper development tools. To give the audience a better understanding of wrappers and the challenges in automatically identifying wrappers, we start the demonstration with the execution.

The demonstration starts with Visual OXPath, a visual IDE for OXPath, shown in Figure 6b and illustrated further in the screencast at `diadem-project.info/oxpath/visual`. With this UI, we



(a) OPAL (b) OXPath

Figure 6: Demonstration UIs

demonstrate how a human would construct a wrapper supported by a visual interface and how a supervised wrapper generator can generalize a wrapper from just a few examples. We provide a number of standard examples, but are also open to suggestions for the audience. Once the wrapper is created we submit it and let it run in the background while we continue with the rest of the demonstration.

In the second part of the demonstration, we focus on the DIADEM prototype. A screencast of the demo is available from `diadem-project.info/screencast/diadem-01-april-2011.mp4`. In the demonstration, we let the prototype run on several pages from the UK real estate and used car domain. The prototype runs in interactive mode, i.e., it visualizes its deliberations and conclusions and prompts the user before continuing to a new page, so that the user can inspect the results on the current page. With this, we can demonstrate first how DIADEM analyses forms and explores web sites based on these analyses. In particular, we show how DIADEM deals with form constraints such as mandatory fields, multi stage forms and other advanced navigation structures. The demo fully automatically explores the web site until it finds result pages for which it performs a selective analysis sufficient to generate the intended wrapper. As before, the demonstration discusses a number of such result pages and shows in the interactive prototype how DIADEM extracts objects from these result pages as well as identifies further exploration steps.

We conclude the demonstration with a brief outlook into applications of DIADEM technology beyond data extraction: The screencast at `diadem-project.info/opal` shows how we use DIADEM's form understanding to provide advanced assistive form filling: where current assistive technologies are limited to simple keyword matching or filling of already encountered forms, DIADEM's technology allows us to fill any form of a domain with values from a master form with high precision.

5. REFERENCES

[1] M. Benedikt, G. Gottlob, and P. Senellart. Determining relevance of accesses at runtime. In *PODS*, 2011.

[2] A. Calì, G. Gottlob, and A. Pieris. New expressive languages for ontological query answering. In *AAAI*, 2011.

[3] A. Calì, G. Gottlob, and A. Pieris. Querying conceptual schemata with expressive equality constraints. In *ER*, 2011.

[4] A. Calì, G. Gottlob, and A. Pieris. Query answering under non-guarded rules in datalog$^{\pm}$. In *RR*, 2010.

[5] V. Crescenzi and G. Mecca. Automatic information extraction from large websites. *J. ACM*, 51(5), 2004.

[6] N. N. Dalvi, R. Kumar, and M. A. Soliman. Automatic wrappers for large scale web extraction. In *VLDB*, 2011.

[7] O. de Moor, G. Gottlob, T. Furche, and A. Sellers, eds. *Datalog Reloaded, Revised Selected Papers*, LNCS. 2011.

[8] E. C. Dragut, T. Kabisch, C. Yu, and U. Leser. A hierarchical approach to model web query interfaces for web source integration. In *VLDB*, 2009.

[9] T. Furche, G. Gottlob, G. Grasso, X. Guo, G. Orsi, and C. Schallhart. Real understanding of real estate forms. In *WIMS*, 2011.

[10] T. Furche, G. Gottlob, G. Grasso, G. Orsi, C. Schallhart, and C. Wang. Little knowledge rules the web: Domain-centric result page extraction. In *RR*, 2011.

[11] T. Furche, G. Gottlob, X. Guo, C. Schallhart, A. Sellers, and C. Wang. How the minotaur turned into Ariadne: Ontologies in web data extraction. *ICWE*, (keynote), 2011.

[12] T. Furche, G. Gottlob, G. Grasso, C. Schallhart, and A. Sellers. OXPath: A language for scalable, memory-efficient data extraction from web applications. In *VLDB*, 2011.

[13] G. Gottlob, T. Lukasiewicz, and G. I. Simari. Answering threshold queries in probabilistic datalog+/- ontologies. In *SUM*, 2011.

[14] G. Gottlob, T. Lukasiewicz, and G. I. Simari. Conjunctive query answering in probabilistic datalog+/- ontologies. In *RR*, 2011.

[15] G. Gottlob, G. Orsi, and A. Pieris. Ontological queries: Rewriting and optimization. In *ICDE*, 2011.

[16] N. Kushmerick. Wrapper induction: efficiency and expressiveness. *AI*, 118, 2000.

[17] G. Orsi and A. Pieris. Optimizing query answering under ontological constraints. In *VLDB*, 2011.

[18] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. Textrunner: open information extraction on the web. In *NAACL* 2007.