DIADEM domain-centric intelligent automated data extraction methodology

DEPARTMENT OF COMPUTER SCIENCE
OXFORD UNIVERSITY

Wolfson Building, Parks Road
Oxford OX1 3QD, U.K.

Andrew Sellers
andrew.sellers@cs.ox.ac.uk

diadem-project.info/oxpath

Registration №:	2011-		
Program title:	OXPath—Open Source, Standard-based Web Data Extraction		
Project leader:	Andrew Sellers	Host Institution:	Chancellor, Masters and Scholars of the University of Oxford

OXPath Overview

OXPath is a scalable, memory-efficient formalism for automatically extracting data from modern web applications. OXPath (a.k.a. *"Oxford XPath"*) is a language for specifying web extraction tasks with a few minimal additions to XPath, the W3C standard for navigation in HTML (and XML) pages. OXPath incorporates the latest web technologies in order to support precise web navigation, interactivity with sophisticated web interfaces, and highly-efficient data extraction. In contrast to many, even commercial data extraction frameworks, it is thus able to extract data from nearly any web site, even those with significant scripting. It is accompanied by a visual interface for easily creating OXPath expressions, that allows anyone slightly familiar with XPath to quickly build wrappers in OXPath. We are currently finalizing the OXPath engine, developing Ox Latin, a Hadoop-based host language for distributed evaluation of OXPath, and extending the visual interface to a full OXPath IDE including visual debugging.

OXPath has originally been developed as part of the DIADEM ERC project at Oxford University, but has since matured into its own project with increasing external collaborators and use beyond Oxford (e.g., for internet archiving at Télécom ParisTech, for data extraction at Vienna Technical University, or for mashup specification at the University of the Basque Country).

Development Goals

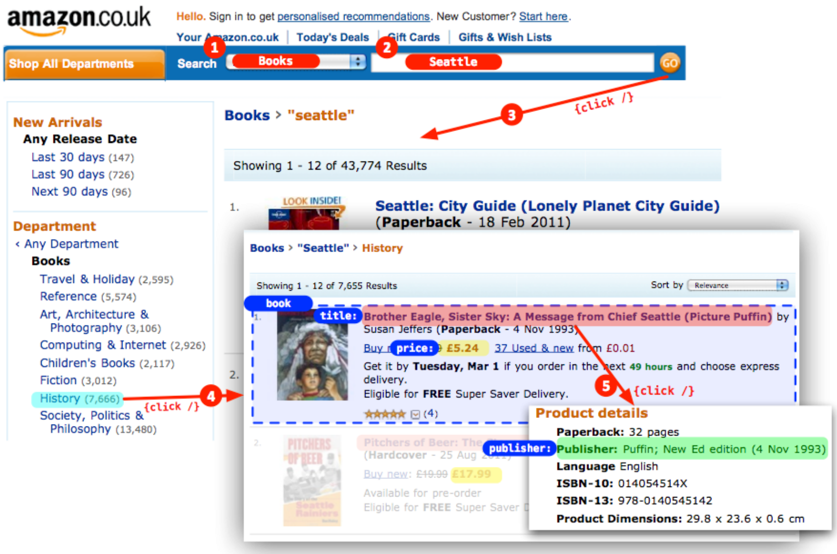
OXPath is a declarative language where a single OXPath expression can populate the fields of a web form, navigate the resulting page(s), and collect the desired information. The *choice of XPath* as basis for OXPath is natural, as it is the standard path language for the XHTML/XML pages found in the world-wide web; as a standard, it has matured in the last few years to now be ubiquitous across the web with APIs in JavaScript and other web languages. Our approach is *robust* enough to compensate for changes in the visual layout of the web content as well as *expressive* enough to capture virtually all web pages in the domains we have investigated.

Data accessible to humans through existing web interfaces needs to be transformed into structured data to be amenable to automated processing. For instance, a gray `span` with class `source` on Google News into the source attribute of that news item. We define the following development goals of OXPath as follows:

1. OXPath can **interact** with rich interfaces of web applications by simulating user actions,
2. OXPath provides extraction capabilities sufficiently **expressive** and precise to specify the data for extraction,
3. OXPath **scales** well even if the number of relevant web sites is very large, and
4. OXPath is **embeddable** in existing programming environments for servers and clients.

The OXPath expression to the left should be familiar to XPath users, but provides additionally functionality for retrieving web pages (via the **doc** function), a new node test for interactive form fields (the *field()* node test), actions (inside curly braces as in {click}), and extraction markers (denoted by :<*book*>).

Processing these expressions efficiently and correctly is the goal of this project. We are

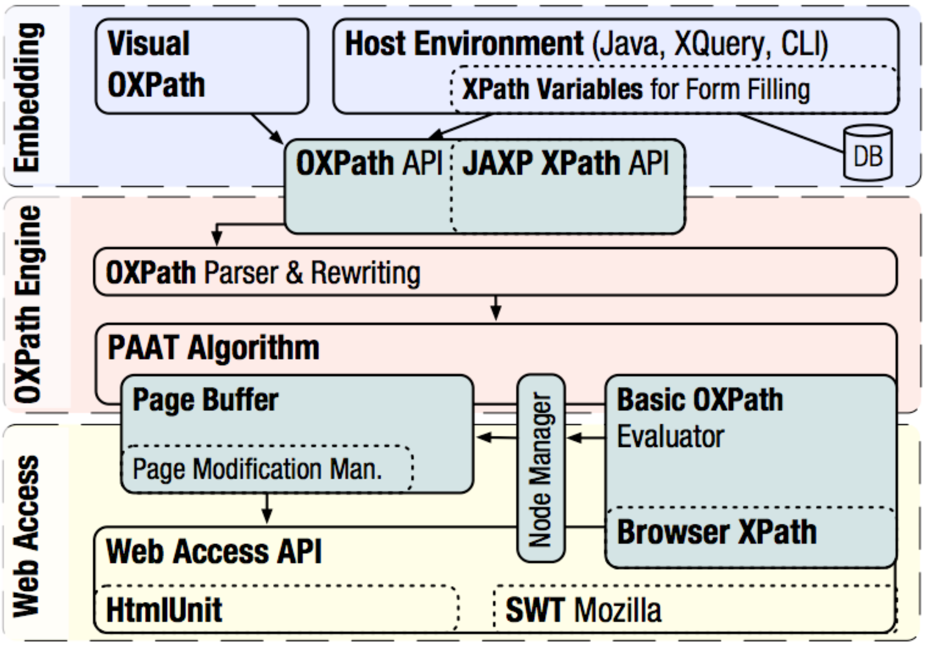

amazon.co.uk
Hello. Sign in to get personalised recommendations. New Customer? Start here.
Your Amazon.co.uk | Today's Deals | Gift Cards | Gifts & Wish Lists
Shop All Departments
Search
Books
Seattle
GO
1
2
3
{click /}
Books > "seattle"
Showing 1 - 12 of 43,774 Results
New Arrivals
Any Release Date
Last 30 days (147)
Last 90 days (726)
Next 90 days (96)
Department
‹ Any Department
Books
Travel & Holiday (2,595)
Reference (5,574)
Art, Architecture & Photography (3,106)
Computing & Internet (2,926)
Children's Books (2,117)
Fiction (3,012)
History (7,666)
Society, Politics & Philosophy (13,480)
4
{click /}
1. LOOK INSIDE!
Seattle: City Guide (Lonely Planet City Guide) (Paperback - 18 Feb 2011)
Books > "Seattle" > History
Showing 1 - 12 of 7,655 Results
Sort by Relevance
book
title:
Brother Eagle, Sister Sky: A Message from Chief Seattle (Picture Puffin) by Susan Jeffers (Paperback - 4 Nov 1993)
Buy new price: £5.24 37 Used & new from £0.01
Get it by Tuesday, Mar 1 if you order in the next 49 hours and choose express delivery.
Eligible for FREE Super Saver Delivery.
★★★★★ (4)
5
{click /}
2. Pitchers of Beer: The ...
(Hardcover - 25 Aug 2011)
Buy new: £19.99 £17.99
Available for pre-order
Eligible for FREE Super Saver D...
Product details
Paperback: 32 pages
publisher:
Publisher: Puffin; New Ed edition (4 Nov 1993)
Language English
ISBN-10: 014054514X
ISBN-13: 978-0140545142
Product Dimensions: 29.8 x 23.6 x 0.6 cm

```
doc("amazon.co.uk")//field()[@title='Search in']/{"Books"}❶
  /following::field()[@title='Search for']/{"Seattle" /}❷
  //field()[@alt='Go']/{click /}❸
  //a[*.refinementLink[.#='History']]/{click /}❹
  //*.result:<book>[.//a.title:<title=(.)>/{click /}❺
    //b[.#='Publisher']/following-sibling::*:<publisher=(.)>]
    [.//span.price[1]:<price=(.)>]
```

developing an engine that supports OXPath evaluation by interfacing with live browsers in order to properly render pages and interact with web applications by simulating user actions. Further, we provide a visual graphical user interface that allows for the development of OXPath expressions, which includes a generation tool that makes suggestions for appropriate subexpressions that best specify extraction tasks.

System Architecture

OXPath uses a three-layer architecture as shown below:



Web Access: In order to programmatically access web pages, OXPath promotes a common interface based on the Document Object Model (DOM) that supports interchangeability of the underlying browser. We currently support browsing environments such as Mozilla SWT and HtmlUnit.

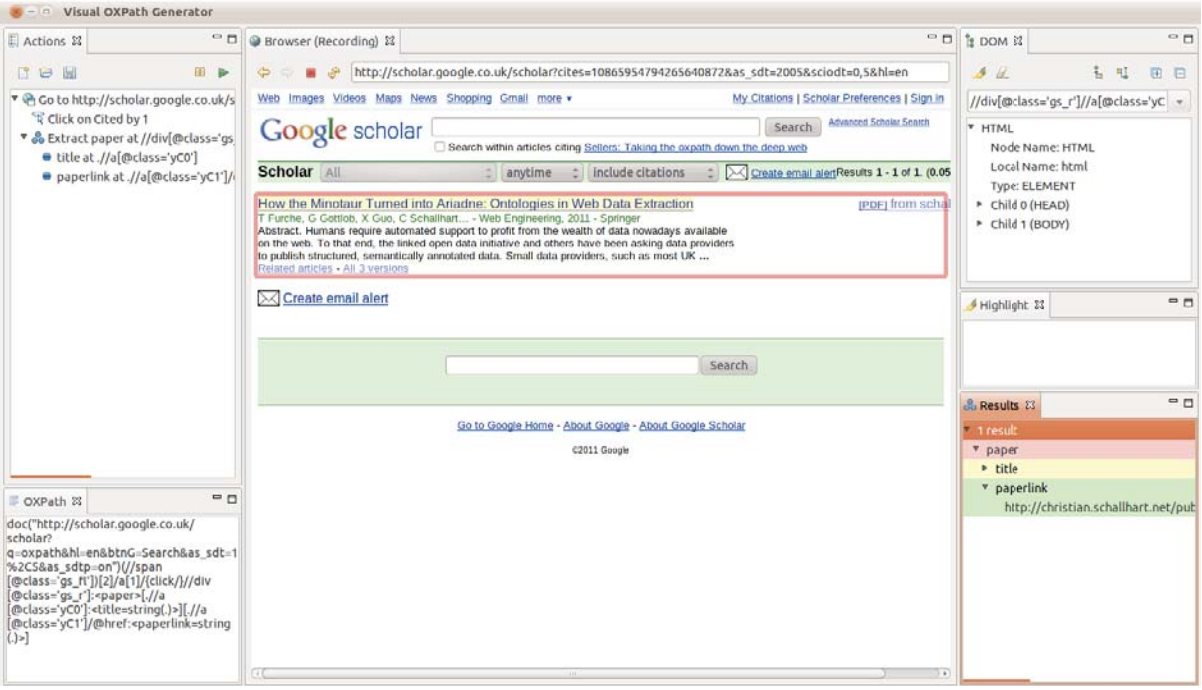
Engine Layer: The OXPath engine is implemented in Java, requiring Java SE 6. This layer efficiently evaluates OXPath expression with our Page-At-A-Time (PAAT) algorithm that guarantees memory use independent of the number of visited pages while remaining polynomial in evaluation time. The current implementation is supported in either Windows 32-bit or Linux 64-bit platforms running Java SE 6.

Embedding Layer: The Visual OXPath tool is also written in Java and built on the Eclipse platform. Further, OXPath is meant to be embeddable into other languages via its API.

Currently, we offer a host language for Java and another implemented via User Defined Functions (UDFs) in Hadoop's Pig Latin to allow distributed OXPath evaluation in a cloud-computing environment.

Visual OXPath

The OXPath engine provides an API to evaluate OXPath expressions programmatically. Visual OXPath (shown below) allows visual selection of navigation and extraction elements via a live browser. Actions can be specified or recorded and generated OXPath expressions, visited DOM nodes, and results are visually reported.



- Go to http://scholar.google.co.uk/s
  - Click on Cited by 1
  - Extract paper at //div[@class='gs
    - title at .//a[@class='yC0']
    - paperlink at .//a[@class='yC1']/

http://scholar.google.co.uk/scholar?cites=10865954794265640872&as_sdt=2005&sciodt=0,5&hl=en

□ Search within articles citing Sellers: Taking the oxpath down the deep web

Results 1 - 1 of 1. (0.05

How the Minotaur Turned into Ariadne: Ontologies in Web Data Extraction [PDF] from schal
T Furche, G Gottlob, X Guo, C Schallhart... - Web Engineering, 2011 - Springer
Abstract. Humans require automated support to profit from the wealth of data nowadays available on the web. To that end, the linked open data initiative and others have been asking data providers to publish structured, semantically annotated data. Small data providers, such as most UK ...
Related articles - All 3 versions

//div[@class='gs_r']//a[@class='yC

- HTML
  - Node Name: HTML
  - Local Name: html
  - Type: ELEMENT
  - Child 0 (HEAD)
  - Child 1 (BODY)

- 1 result
  - paper
    - title
    - paperlink
      - http://christian.schallhart.net/pub

```
doc("http://scholar.google.co.uk/
scholar?
q=oxpath&hl=en&btnG=Search&as_sdt=1
%2C5&as_sdtp=on")(//span
[@class='gs_ft'])[2]/a[1]/{click/}//div
[@class='gs_r']:<paper>[.//a
[@class='yC0']:<title=string(.)>][.//a
[@class='yC1']/@href:<paperlink=string
(.)>]
```

Development Language

OXPath is written in Java and is developed with several open-source Java tools, including Eclipse, JavaCC, and HtmlUnit. The cloud-based host language additionally uses Pig Latin and Hadoop, which are also Java open source tools.

Systems & Components used

OXPath was developed with Java using Eclipse on Linux, Windows, and Mac OS X computers. OXPath could not exist without the numerous open source tools developed by the Java and web engineering community. First and foremost, we embed either Mozilla or Qt Webkit as browser engine. To access these engines from Java, we provide a unified Java browser API that encapsulates details of the underlying APIs (for Mozilla: JavaXPCom and SWT, for Qt a heavily modified Qt Webkit API translated using Jambi). Second, we use the Apache Hadoop framework for the cloud version of OXPath and its host language Ox Latin. Third, we use the Eclipse plugin framework for the visual interface and the planned OXPath IDE.

Development Plan

Milestone	Time
Develop **core** language/**PAAT** algorithm, prototyping	Aug—Dec 2010
Develop **OXPath 0.8** (based on HTMLUnit)	Jan—Apr 2011
Develop browser framework for extraction from scripted pages	Sep 2010-May 2011
Experimental evaluation, testing	May 2011
Migrate OXPath to browser framework (version 0.9)	Jul 2011
Automated test case development/testing	Aug 2011
Design and development of visual OXPath for visual generation	May-Sep 2011
Development of **Ox Latin**, cloud-based host language	Jul-Sep 2011
Finalize OXPath **version 1.0**	End of Aug 2011
Integrate, prepare documentation	Middle of Sep 2011
Integrate visual debugger	Oct 2011
Improve and test support for WebKit	Oct/Nov 2011
OXPath Suite 1.0 (Engine, developer, editor, debugger)	Dec 2011

Project Team & Roles

Project lead:	Andrew Sellers	Core engine:	Andrew Sellers
Language:	Tim Furche, Georg Gottlob, Giovanni Grasso, Christian Schallhart, Andrew Sellers		
Visual OXPath:	Jochen Kranzdorf	Ox Latin (Cloud):	Giovanni Grasso, Andrew Sellers
Integration:	Andrew Sellers	Testing:	Christian Schallhart, Andrew Sellers

External collaborators: Qianze Zhang, Michael Benedikt, Pierre Senellart

Long-term Prospects of OXPath

OXPath has already shown great promise as a web data extraction platform. To the best of our knowledge, OXPath is the first web extraction system with strict memory guarantees, which reflect strongly in our experimental evaluation which OXPath to other systems. We believe that it can become an important part of the toolset of developers interacting with the web. It is also the **only open source data extraction platform** that is able to deal with modern web sites. Other open source frameworks, such as WebHarvest, treat the web as more or less static HTML pages with no need to render web pages or execute Javascript. In our experiments, this assumption does no longer hold for most pages. Even HtmlUnit that tries to simulate a browser for web automation, falls short on a significant number of web sites, in particular where scripting or AJAX are involved. The OXPath approach, to operate on top of a browser API that encapsulates several modern browser APIs, has already proven far more robust and is likely to be easy to maintain, as browsers evolve.

We are committed to building a strong set of tools around OXPath. We provide a visual generator for OXPath expressions and a Java API. Some of the issues raised by OXPath that we plan to address in future work are: (1) OXPath is amenable to significant optimization and a good target for automated generation of web extraction programs. (2) Further, OXPath is perfectly suited for highly parallel execution: Different bindings for the same variable can be filled into forms in parallel. Toward this end, we are currently developing a host language for OXPath that can effectively parallelize OXPath expression evaluation while gaining aggregation, query composition, and integration capabilities. (3) We plan to further investigate language features, such as more expressive visual features and multi-property axes.

As detailed above, we plan to release version 1.0 of OXPath in September and a first version of the full OXPath IDE in December/January. Development will continue after that in Oxford, as OXPath is a core activity of the DIADEM project and figures prominently into the DIADEM system of data extraction. However, we also envision increasing participation by external collaborators. Already OXPath is being used in several other research groups all over the Europe (Vienna, Paris, and Bilbao). The interest in OXPath is further demonstrated by a recent Amazon grant supporting the cloud version of OXPath.

Furthermore, we intend to spin of several components of OXPath and DIADEM as open source projects of there own, most prominently the Java web browser API that finally gives a working, open source, browser-independent solution for browser access and embedding in Java APIs and the framework for recording and visual highlighting in a browser that is used in visual OXPath, but has a broader applicability, e.g., for ontology population or automation and testing.