Semantic Markup in Web Data Extraction

A thesis submitted to the

University of Dublin, Trinity College

for the degree of

Master of Science

Tewson Seeoun

Knowledge and Data Engineering Group

School of Computer Science and Statistics

Trinity College Dublin

Ireland

2013

DECLARATION

I, the undersigned, declare that this work has not been previously submitted as an exercise for a degree at this or any other university, and that, unless otherwise stated, it is entirely my own work.

Tewson Seeoun

April 2013

PERMISSION TO LEND OR COPY

I, the undersigned, agree that the Trinity College Library may lend or copy this thesis upon request.

Tewson Seeoun

April 2013

ACKNOWLEDGEMENTS

I would like to express my deepest and most sincere gratitude to my supervisor Dr. Declan O'Sullivan and my co-supervisor Dr. Rob Brennan. I am forever indebted to their kind guidance, support and amazing patience. I have learnt a great deal from them.

I would like to also thank the participants in my experiment. Without them this thesis would not have been possible. I would like to thank people in KDEG for their friendship and advice.

Finally, I would like to acknowledge the Networked and Embedded Systems project (NEMBES), the Telecommunications Graduate Initiative (TGI), the Federated, Autonomic Management of End-to-End Communication Services Strategic Research Cluster (FAME) and Trinity College Dublin for funding this research.

ABSTRACT

A recent approach to publishing data on the web called *semantic markup* has enabled web pages to be annotated with machine-readable linked data. Semantic markup allows publishers to associate and describe HTML elements on a web page with existing and usually commonly used metadata vocabularies. However, there has been little acknowledgment of semantic markup in state-of-the-art web data extraction techniques. This thesis investigates the possibilities of using semantic markup data in web data extraction, specifically as a method of data identification. This thesis also investigates the use of semantic markup data on a web page as an initial step to extract more data from surrounding HTML elements. This thesis proposes a process of using semantic markup data in web data extraction called the *external model enrichment process*. A web data extraction tool called the semantic markup mapping tool was created to explore the use of semantic markup in web data extraction. A user experiment was conducted to evaluate the semantic markup mapping tool. The external model enrichment process was also implemented and evaluated. The evaluation of the external model enrichment process provided evidence that precision and recall of web data extraction could be improved with the use of semantic markup data. Further possibilities of improvement of the external model enrichment process have also been documented.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

This chapter introduces the thesis. First it presents the motivation, followed by the central research question and research objectives. Then summaries are presented of the area of contribution, the implementation artefacts, the evaluation strategy used in the research and finally key points from each chapter from the rest of the thesis.

1.1 MOTIVATION

Data is published and consumed constantly through the World Wide Web (WWW). There is machine-readable data that has been designed to be processed by computer programs. There is also data aimed towards human consumers such as the actual information on web pages. To enable computer programs to process data that was not originally designed to be machine readable, a research area called *web data extraction* has been developed (Laender et al. 2002).

A web data extraction process consists of two key functions: data identification and mapping (Laender et al. 2002). During data identification, algorithms are used to identify the data of interest in a data source. The identified data is then mapped to a data structure that is suitable for the desired applications.

A recent trend in publishing data on the web is to annotate HyperText Markup Language (HTML) elements with linked data (Mühleisen and Bizer 2012). This method of annotating HTML elements with linked data is referred to in this thesis as *semantic markup*[1]. Semantic markup allows publishers to “mark” parts of the data on a web page for a computer program to identify. Current uses of semantic markup include search engine optimisation and content personalisation (Steiner et al. 2010, Plumbaum et al. 2012).

Since semantic markup allows computer programs to identify data, it could be used in the data identification function of a web data extraction process. In this way it may be used as an alternative approach to the traditional web data extraction techniques.

[1] A detailed description of semantic markup can be found in section 2.4.

Traditional web data extraction techniques use HTML source code, and oftentimes the World Wide Web Consortium (W3C) standard Document Object Model (DOM)[2] representation of the source code, as inputs. Since semantic markup data is associated with HTML elements in the DOM, the data identification process could be extended from the annotated HTML elements to the surrounding DOM data. In this way more data can be extracted from the web page.

However, to the author's knowledge, no researchers have yet exploited this method of using semantic markup data and related DOM data for web data extraction. The method could help assist in the data identification step and the mapping of the identified data.

This thesis presents research on the use of semantic markup data and the DOM in the web data extraction process. The next section presents the research question underlying the thesis.

1.2 Research Question

This thesis investigates the following question:

"To what extent can the W3C document object model (DOM), including embedded semantic markup, assist in identifying and mapping data in a web data extraction process?"

1.3 Research Objectives

This thesis has five research objectives derived from the research question. These are listed below:

- RO1: To conduct a survey on state-of-the-art research on web data extraction, applications of semantic markup, and semantic mapping.
- RO2: To design and implement a semantic markup mapping tool that exploits the semantic markup's links to HTML elements.
- RO3: To conduct an evaluation of the semantic markup mapping tool.

[2] W3C Document Object Model, http://www.w3.org/DOM/, Last visited: September 2012.

- RO4: To design and implement a process for extracting web data using semantic markup and the DOM.
- RO5: To conduct an evaluation on the process for extracting web data using semantic markup and the DOM.

1.4 CONTRIBUTION

A major contribution of this thesis is the proposed process of web data extraction using semantic markup accompanied with data from the DOM. This process, discussed in detail in chapter 7, is called the ***external model enrichment process***. The popularity of semantic markup has been growing. Meanwhile, a survey on state-of-the-art research shows that there has been little research on using semantic markup in web data extraction.

An experiment was conducted to evaluate the external model enrichment process. In the experiment, two sets of results from the mapping part of a web data extraction process are compared. One set of results was created from using only semantic markup data. Another set of results was created using the proposed enrichment process. The results achieved using the enrichment process at this stage have not yet shown much increase in precision and recall compared to using semantic markup data alone. However, the experiment results and analysis have shown that there is definite potential in improving the current web data extraction process using the proposed external model enrichment process.

A minor contribution of this thesis is the design of the semantic markup mapping tool. The semantic markup mapping tool was designed and implemented in an earlier phase of this thesis. The tool was designed to explore the possibilities of using semantic markup in web data extraction. The tool highlights HTML elements annotated with semantic markup data in a manual semantic mapping process. A user experiment on the tool was conducted with participants with different levels of expertise in semantic web technologies. The tool proved to be useful to participants who are most familiar with semantic web technologies. The results from the user experiment also showed that a major part of the mistakes users make are trivial and could be eliminated with more training.

Another minor contribution of this thesis is the software prototypes that have been implemented. Two Google Chrome web browser extensions have been created for evaluations. One was created to evaluate the design of the semantic markup mapping tool. Another was created to evaluate the external model enrichment process. Both tools were developed with the HyperText Markup Language and the JavaScript language which are very popular in web development. In the author's opinion, this makes the software prototypes easy to be learned from and developed further.

1.5 Technical Approach

After the research question was established, a survey on state-of-the-art research was conducted on the related areas. The survey was conducted first on the current applications of semantic markup data. This was to find out whether semantic markup was used in web data extraction. The survey shows that as yet there has not been much use of semantic markup in web data extraction, with one notable work discussed in section 3.2.2. After that, the survey was conducted on current techniques in web data extraction. This was to verify if semantic markup data was listed as an input to a web data extraction technique. The survey shows that current web data extraction techniques still use only HTML source code and its DOM representation.

Following the survey, the semantic markup mapping tool was designed to investigate the use of semantic markup in web data extraction. A web browser extension was created as a software prototype of the semantic markup mapping tool. The software prototype was used in an experiment to evaluate the semantic markup mapping tool. The evaluation showed that the association between semantic markup data and annotated HTML elements is perceived to be useful by the users.

After the semantic markup mapping tool, the external model enrichment process was designed. The external model enrichment process uses a set of pre-defined strategies to add RDF properties to the external model based on data in the DOM. A Google Chrome web browser extension was created to implement the enrichment process. An experiment was conducted to evaluate the enrichment process by comparing precision and recall in web data extraction with and without the enrichment process.

From both experiments, a conclusion is drawn and reflection upon the research objectives. Future work is also documented.

1.6 Evaluation Strategy

This section describes strategies for evaluating the two key artefacts proposed in the thesis.

The first experiment was conducted to evaluate the semantic markup mapping tool. The experiment was designed to be a user experiment. The experiment was designed to measure and compare accuracy and speed achieved in using the tool by participants with different levels of expertise. The experiment also had a questionnaire enquiring the participants on perceived usability of the tool and criticism on the tool's features.

The second experiment was conducted to evaluate the external model enrichment process in web data extraction. In the experiment, results were obtained from two approaches: the baseline approach and the experimental approach. The baseline approach yielded web data extraction results without using the external model enrichment process. The experimental approach yielded results using the external model enrichment process. Both sets of results are compared against an ideal set of web data extraction results called the gold standard. Precision and recall scores are compared between the two approaches.

1.7 Thesis Overview

In this section an abstract of each of the remaining chapters in the thesis is presented.

Chapter 2 presents background knowledge on technologies related to this thesis. This chapter begins with the development of web data extraction as a research area. Following that, the mechanism of how web pages are published and displayed is explained. The next section introduces the concept of linked data, which is an approach to publishing structured data on the web. After that, the concept of semantic markup is presented. Finally, the last section introduces a data integration process called semantic mapping. Current semantic mapping tools, both manual and automated, are presented. CogZ is chosen as an inspiration on manual mapping and the Alignment API is chosen for automated mapping

Chapter 3 provides a survey on state-of-the-art research in semantic markup, web data extraction and semantic mapping. Current applications of semantic markup are search engine optimisation, client-side display augmentation for domain-specific data, and web content personalisation. Semantic markup is also found to be used in web data extraction, but its association to HTML elements is not used. Current web data extraction techniques use HTML source code and its document object model representation as inputs. Recent surveys on web data extraction do not mention semantic markup and its association to HTML elements as inputs.

Chapter 4 presents the generic web data extraction process used in this thesis. The process begins by using semantic markup data to identify relevant data. An RDF graph is constructed from the identified data and is called the external model. The target model for web data extraction is called the internal model. The external model and the internal model are then mapped using a semantic mapping tool of choice.

Chapter 5 presents a design and implementation of a semantic markup mapping tool. The semantic markup mapping tool is implemented from a set of use cases derived from CogZ's cognitive support framework. The tool uses semantic markup to identify data on a web page and uses a manual mapping interface to let the user generate mapping correspondences. The tool is implemented as a Google Chrome web browser extension.

Chapter 6 presents an evaluation of the semantic markup mapping tool from chapter 5. The evaluation was conducted with the hypothesis that participants of different expertise would use the tool with similar speed and accuracy. The results show that the group of domain experts still perform better than the rest. However, the results also suggest that more training and alteration to the user interface may help improve accuracy for other user groups.

Chapter 7 presents a design and implementation of a process of adding additional properties from the DOM to the external model. The process is called the *external model enrichment process* and is aimed to increase precision and recall in web data extraction as compared to using only semantic markup data. Four enrichment strategies have been developed based on observation of the DOM. A software prototype was implemented as a Google Chrome web browser extension.

Chapter 8 presents an evaluation of the external model enrichment process. The evaluation was conducted with the hypothesis that a web data extraction process that uses the external model enrichment process would achieve higher precision and recall than the process that does not. The results show that there is little difference in precision and recall. However, mapping correspondences generated using the external model enrichment process have higher confidence values. This is likely due to the fact that added properties from the external model enrichment process increase the probability of the semantic mapping tool to find better correspondences.

Chapter 9 presents the conclusions and future work arising from the research.

2 BACKGROUND

This chapter provides background knowledge on technologies related to the thesis: web data extraction, the HyperText Markup Language (HTML), linked data, semantic markup and semantic mapping. It also provides the terminology that will be used throughout the thesis.

2.1 WEB DATA EXTRACTION

Not all data on the web is immediately machine-readable. There is much data that is readable by human but requires some processing before its structure can be identified by computer programs. Such processing is called *web data extraction*.

A web data extraction system has been defined as "a software system that automatically and repeatedly extracts data from web pages with changing content and delivers the extracted data to a database or some other application" (Baumgartner et al. 2009). In the literature, the terms *web data extraction*, *web information extraction*, *web scraping* and *wrapper generation* are used interchangeably (Baumgartner et al. 2009).

In the late 1990s, the amount of data published on the web was rapidly increasing (Embley et al. 1999). Most of the data published on web pages at that time was seen as being semi-structured. This means while different parts of the data could be distinguished, its structure was not as strict as that of relational databases (Abiteboul 1997). There was much research undertaken on querying and manipulating web data as was undertaken for databases (Laender et al. 2002). Most of the work used the approach of creating a *wrapper* program (Eikvil 1999). Wrappers act as an intermediate between web pages and the user. They "identify data of interest and map them to some suitable format" (Laender et al. 2002). In this way the identified or extracted data would be in a structure or a format that is usable to existing applications. Thus there are two key steps in web data extraction: **identification** of data of interest and **mapping** that data to a known structure.

Use cases of web data extraction systems include web page monitoring for business intelligence (Baumgartner et al. 2005) and data integration (Wong and Hong 2007). Many techniques for identifying relevant data have been developed and will be discussed in section 3.3. In this thesis, a recent method of publishing data in web pages called semantic markup is the topic of interest. The author of the thesis sees semantic markup as an opportunity in identifying data and can be used in web data extraction. The concept of semantic markup is described in section 2.4.

As will be discussed later on, semantic markup data uses a data format called the Resource Description Framework (RDF) (Lassila and Swick 1998). The research area on mapping semantic data published as RDF is called *semantic mapping*. The concept of semantic mapping is described in section 2.5.

Before discussing data publication in web pages with semantic markup, the mechanism of how web pages are created and processed is described in the following section.

2.2 Web Pages

This section describes how a web page is created. This thesis is focused only on web pages created with HTML. First, the concept of markup languages and HTML in particular is described in sub-section 2.2.1. Then, the mechanism of web page rendering – the process of reading HTML code and displaying a web page on a screen – is described in sub-section 2.2.2.

2.2.1 HTML and Other Markup Languages of the Web

A markup language is a system of *marking up* or adding structure to a text document with distinct strings of text tokens. These strings of text are called tags (Coombs et al. 1987). Adding structure to a document with tags allows identifying different parts of the document. For example, consider the following code snippet:

```
:h1.A Survey on Markup Languages
:p.This paper presents a survey on markup languages.
```

The code snippet is written in the Generalized Markup Language (GML)[3]. GML was developed by the technology company IBM in the 1960s as a method to format text. GML tags are defined as strings of text that are preceded with a colon and followed by a dot. This

[3] GML Starter Set User's Guide, http://publibfp.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/dsm04m00/CCONTENTS, Last visited: January 2013.

way, the first line of the code can be interpreted as being formatted by the tag *h1*, which represents a document header[4]. The tag *p* on the second line represents a paragraph.

Over time, new markup languages have been developed. The rest of this sub-section describes the markup languages involved in this thesis. The markup languages are introduced in this order: the HyperText Markup Language (HTML), the Extensible Markup Language (XML), the Extensible HyperText Markup Language (XHTML) and, finally, the latest development of the HyperText Markup Language called HTML5.

HTML

The HyperText Markup Language (HTML)[5] is a markup language for creating web pages. HTML has a set of standard tags that are used to create elements to assemble a web page. In HTML, tags are identified by being enclosed in the signs < and >. Some tags can be used on their own. The following example is a tag used to break a line to start a new one:

```
<br>
```

Some tags can be used to enclose a string of text with opening and closing tags. The sign / is used to indicate a closing tag. For example, marking up a title of a document can be written as:

```
<title>A Survey on Markup Languages</title>
```

A complete single tag or a complete set of opening and closing tags forms an *element*. Additional data for each HTML element can be provided through strings of text within the tag. These strings of text are parsed by the web browser as sets of key-value data called *attributes*. For example, to assign a unique identifier to a paragraph, an ID attribute can be used. The following code snippet demonstrates the use of the ID attribute in a paragraph element:

```
<p id = "paragraph01">This is an example paragraph.</p>
```

[4] GML Starter Set Reference, http://publibfp.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/dsm05m00/CCONTENTS, Last visit: January 2013.

[5] W3C HTML, http://www.w3.org/html/, Last visited: September 2012.

Authors can control the appearances of HTML elements with an attribute called *class*. The class attribute is usually used to refer to a set of style commands called a Cascading Style Sheet (CSS). CSS and appearances of HTML elements will be discussed further in sub-section 2.2.2.

XML

The Extensible Markup Language (XML)[6] is another language that is widely used to publish data on the web. XML also uses tags to add structure to the data. However, authors can create their own tags. Unlike HTML, XML was not intended to be displayed as web pages. It is used more as a format for data exchange. XML also has stricter validation rules than HTML, usually referred to as *well-formedness* of the document.

XHTML

The Extensible HyperText Markup Language (XHTML)[7] was developed as a variety of XML for web pages. In other words, it is XML with a list of pre-defined tags taken from HTML. In this way, web pages published in XHTML can also be processed by XML applications. The XHTML standard is intended to ensure XHTML web pages to be well-formed. This is achieved by getting web browsers to stop rendering web pages that are not well-formed and instead display an error message. This strict error handling is colloquially referred to as being “Draconian”[8]. HTML parsing which allows non well-formed documents is considered more “forgiving” than XHTML. To encourage publishers in transition from HTML to XHTML, web pages written in XHTML 1.0 are allowed to be parsed as HTML. However, in XHTML 1.1, well-formedness is strictly enforced[9].

[6] Extensible Markup Language (XML), http://www.w3.org/XML/, Last visited: September 2012.

[7] XHTML 1.0 The Extensible HyperText Markup Language (Second Edition), http://www.w3.org/TR/xhtml1/, Last visited: September 2012.

[8] Draconian Error Handling, http://www.w3.org/html/wg/wiki/DraconianErrorHandling, Last visited: January 2013.

[9] XHTML 1.1 - Module-based XHTML - Second Edition, http://www.w3.org/TR/xhtml11/, Last visited: September 2012.

HTML5

Since XHTML has stricter rules, parsing errors are more likely to be displayed. For example, it is likely to be more difficult to ensure well-formedness of a web page which is generated and assembled programmatically from different parts of a website. To address this issue, a group from web technology companies called The Web Hypertext Application Technology Working Group (WHATWG)[10] were formed. WHATWG attempted to revise the HTML specification that would introduce new technologies and still be backward compatible to existing HTML web pages. This resulted in a new version of HTML, HTML5[11]. HTML5 has not yet been declared by W3C as a standard. However, the specification drafts of HTML5 have been published by W3C. W3C have also announced[12] the intention to make HTML5 a W3C standard by 2014. HTML5 introduces new application programming interfaces (APIs) and HTML elements such as local data storage[13] and programmable graphics[14].

2.2.2 Web Page Rendering

Computer programs called *web browsers* use HTML documents containing HTML source code to create web pages and display them on a computer screen. Examples of commonly used modern web browsers are Microsoft Internet Explorer[15], Mozilla Firefox[16] and Google Chrome[17].

[10] Web Hypertext Application Technology Working Group, http://www.whatwg.org, Last visited: September 2012.

[11] HTML5, http://www.w3.org/TR/html5, Last visited: September 2012.

[12] W3C confirms May 2011 for HTML last call, targets 2014 for HTML5 standard, http://www.w3.org/2011/02/htmlwg-pr.html, Last visited: March 2013.

[13] HTML5 allows websites to store data in web browsers for client-side use.

[14] HTML5 introduces a new element called *canvas*. Canvas elements are used as a container of programmatically generated graphics. The graphics are created using another markup language called the Scalable Vector Graphics language (http://www.w3.org/TR/SVG, Last visited: March 2013).

[15] Internet Explorer, http://www.microsoft.com/InternetExplorer, Last visited: January 2013.

[16] Mozilla Firefox Web Browser, http://www.mozilla.org/firefox, Last visited: January 2013.

[17] Chrome Browser, http://www.google.com/chrome, Last visited: January 2013.

Appearances of an HTML element are indicated by a set of key-value data called *styles*. Styles may be defined in an HTML document or in an external document called a Cascading Style Sheets (CSS) document[18]. Web browsers have default CSS documents[19]. Both default and external CSS are used by web browsers through a *rendering engine*[20] to calculate how HTML elements are displayed. For example, the rendering engine would display text in a relatively larger size by default if the text is enclosed in the tags <h1></h1>. The font type of that text may be specified in an external CSS document.

A style value can be specified in an attribute. For example, the attribute *border* defines values for border-related styles such as border colour or thickness. However, not all attributes define values for styles. There are attributes are used only to provide data to the web browser and are not rendered onto the screen. Examples of such attributes are *class* and *id*.

2.3 Linked Data

Since 2007, an approach to publishing data on the web called *Linked Data* has been gaining momentum. In a paper that observes the progress of linked data (Bizer et al. 2009), linked data is defined as “data published on the Web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external data sets, and can in turn be linked to from external data sets”.

The web of data before linked data was constructed on linking documents through hypertext links (Berners-Lee 2006). Linked data was proposed to realise a scenario that “all published data becomes part of a single global data space” (Bizer et al. 2009). This is achieved by allowing data to use references to other data. A set of rules for achieving a web of linked data has been proposed as follows (Berners-Lee 2006):

[18] Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification, http://www.w3.org/TR/CSS21/, Last visited: January 2013.

[19] For example, Mozilla Firefox’s default style sheets can be found at http://hg.mozilla.org/mozilla-central/file/tip/layout/style/html.css.

[20] A rendering engine is also called a web browser engine or a layout engine.

- Entities in a linked data document have to be identifiable. It was proposed that those entities are identified by a sequence of characters call a uniform resource identifier (URI).
- Such URIs have to use the HyperText Transfer Protocol (HTTP) as the means of communication.
- To provide information from a URI, it is recommended to use a standardised format such as the Resource Description Framework (RDF).
- A published linked data document should refer to other documents through URIs to encourage information discovery.

RDF (Lassila and Swick 1998) is the dominant format for publishing linked data (Bizer et al. 2009). RDF was designed to be used for describing *resources* or things that can be identified. RDF resources are identified by unique identifiers called the Uniformed Resource Identifiers (URIs) (Berners-Lee et al. 1998). URIs are strings of characters that contain information on hierarchical location of things. One example of a URI is the Uniformed Resource Locator (URL), which is also commonly known as the web address.

With RDF, resources are described in simple *statements*. These statements consist of a subject resource, a predicate or a property, and an object resource. A set of three components that make up a statement is also called a *triple*. Multiple triples form a *graph* in which nodes are resources and edges are properties. An RDF graph can be used to describe relationships between resources where an object of one statement is a subject of another.

RDF graphs can be *serialised* or published in a format that can be stored and retrieved later. The most common RDF serialisation formats are XML or RDF/XML, Notation 3, and Turtle (Segaran et al. 2009). In this thesis, RDF graphs are serialised with the RDF/XML format[21]. This is because the RDF/XML format can be processed by related applications such as web browsers and semantic mapping tools.

[21] RDF/XML syntax specification, http://www.w3.org/TR/rdf-syntax-grammar/, Last visited: January 2013.

RDF can be used to publish *vocabularies*[22]. Vocabularies are sets of terms and their definitions that are used to describe concepts. Organisations publish vocabularies to have a set of agreed definitions in describing concepts. For example, the Friend of a Friend project (FOAF)[23] created a vocabulary on social networks. The FOAF vocabulary contains definitions of social network-related terms such as the class *Person* for describing persons, or the property *workplaceHomepage* for relating a person and their workplace's web page.

An RDF graph may be published using terms from other vocabularies. For example, the Schema.org vocabulary[24] covers a range of topics from recipes to films. However, the Internet Movie Database (IMDb)[25] use only the terms related to films to describe their film entries. From this example, each IMDb film entry described with terms from Schema.org is referred to in this thesis as *instances*. In this thesis, a resource described with a class and its properties defined in a vocabulary is referred to as an instance of that class.

This thesis is focused on RDF graphs that use vocabularies to describe instances. In other words, these RDF graphs describe how their instances are known by using a model. Such a model is composed of terms defined in vocabularies. These RDF graphs are referred to in this thesis as *knowledge models*.

Knowledge models can be found as a serialised RDF graph containing a set of instances. For example, the DBpedia project[26] extracts data from Wikipedia[27], an online encyclopaedia, and publishes the data as RDF graphs. Knowledge models can also be found embedded in web pages where components of the web page are annotated with terms. This method of annotating a web page with RDF terms is discussed in the following section.

[22] Ontologies, http://www.w3.org/standards/semanticweb/ontology, Last visited: January 2013.

[23] The Friend of a Friend (FOAF) project, http://www.foaf-project.org/, Last visited: January 2013.

[24] What is Schema.org?, http://schema.org/, Last visited: September 2012.

[25] IMDb – Movies, TV and Celebrities, http://www.imdb.com/, Last visited: September 2012.

[26] DBpedia, http://dbpedia.org, Last visited: September 2012.

[27] Wikipedia, http://www.wikipedia.org, Last visited: September 2012.

2.4 Semantic Markup

In this work, the term *semantic markup* refers to a method of annotating a web page with semantics which is in the form of linked data. There are now three major semantic markup specifications (Mika and Potter 2012): Microformats (Khare and Çelik 2006), RDFa (Adida et al. 2008) and Microdata (Hickson 2011). This section begins with the motivation behind the development of semantic markup. The rest of this section describes how the three specifications have been developed and deployed.

2.4.1 Embedding Semantics in Web Pages

The W3C reference on HTML tag definitions[28] provides “semantics” for every HTML tag. However, in the opinion of the author of the thesis, this type of semantics is different from semantics used in linked data. The author believes that semantics defined in the W3C reference is what the HTML tags *are*. For example, the tag <p> is a *paragraph* or the tag <a> is a *link*. This semantics is different from the linked data semantics. For example, with linked data, a resource can be described as being a *recipe*. This may be done by stating that the resource URI has a property called *type* which has a value as the recipe class URI. In comparison, the semantics of HTML tag definitions does not include a standardised way to describe an *article* as being a *recipe*.

To avoid ambiguity in using the term *semantics*, in this thesis, the term *semantics* means *linked data semantics*.

There have been attempts to extend HTML to support such semantics. One of the earliest was the Simple HTML Ontology Extensions (SHOE) (Heflin et al. 1999). The SHOE project introduced a set of new tags to HTML. These tags allow the declaration of an ontology – a knowledge model describing concepts of interest and relationships among them – within an HTML document. SHOE also allows embedding instances of concepts from that ontology in the document.

[28] HTML: The Markup Language (an HTML language reference), http://www.w3.org/TR/html-markup/, Last visited: February 2013.

There are also efforts to include semantics in HTML that avoid introducing new tags or attributes to HTML. An example is the Embedded RDF specification or eRDF[29]. It exploits the class attribute in HTML as a reference to a part of a knowledge model.

The following sub-sections present the three most prevalent modern approaches to embedding semantics in HTML documents. The three approaches consist of Microformats, RDFa and Microdata.

2.4.2 MICROFORMATS

A similar effort to eRDF is the Microformat specification or Microformats (Khare and Çelik 2006). It was developed by the Technorati online community[30]. The Microformat specification uses HTML's “class” and “rel” (short for relationship) attributes to specify richer semantics. The specification provides an alternative way to natural language processing for a program to identify data on a web page. For example, a <span> element may contain a *telephone number*. With Microformats, the web page author may assign the class value of “tel” to this <span> element. A program that is aware of Microformats may not have to recognise telephone number formats. The program may instead identify a telephone number on a web page by finding an HTML element with the class “tel”.

With Microformats, the class attribute is used to assign a data type to an HTML element and the rel attribute is used on hyperlinks to define the relationship between documents[31]. The Technorati community have published Microformat data type specifications for domains of data that are common on the web. For example, contact information can be described with the hCard specification[32], example class values of which include *fn* for names and *org* for organisation names.

[29] http://blog.iandavis.com/2005/10/19/introducing-embedded-rdf/

[30] Technorati, http://technorati.com, Last visited: February 2013.

[31] Predefined values of the rel attribute existed before Microformats. However, the community created additional ones such as *nofollow* and *tag*. The rel value *nofollow* forbids search engines to index the linked web page. The rel value *tag* indicates that the link leads to a web page on a category described in the link text.

[32] hCard 1.0, http://microformats.org/wiki/hcard, Last visited: September 2012.

2.4.3 RDFa

As mentioned in section 2.3, one of the most common formats of RDF serialisation is XML. Therefore, a way to publish linked data in the form of RDF in XHTML documents has been developed. It is called RDF-in-attribute (RDFa) (Adida et al. 2008). RDFa uses existing XHTML attributes and also introduces new attributes for publishing RDF triples with an XHTML document. The full list of RDFa attributes can be found at the RDFa specification document[33]. A brief example of using RDFa is shown in the following code snippet.

```
<div about=”#publication01”>
  <p>Title: <span property=”hasTitle”>What is RDFa?</span></p>
</div>
```

The example code contains a subject, a property and an object for a triple. The subject is a fragment of the web page referred to as “publication01”. The property is “hasTitle”. The object is a string of text with the value “What is RDFa?” This triple can be serialised in RDF/XML as the following code snippet.

```
<rdf:Description rdf:about=”#publication01”>
  <hasTitle>
    What is RDFa?
  </hasTitle>
</rdf:Description>
```

If an existing vocabulary is used, terms can be referred to either with their full URIs or the compact version of the URIs called Compact URI (CURIE)[34]. CURIEs are shortened by the use of *prefixes*. A prefix is written as a string of text followed by a colon followed by the desired fragment. For example, in the social network vocabulary Friend of a Friend (FOAF), the full URI of the property *name* is:

```
http://xmlns.com/foaf/0.1/name
```

The prefix of the FOAF vocabulary is “foaf”. If the prefix is used, the URI above can be shortened as `foaf:name`.

[33] RDFa Core 1.1, http://www.w3.org/TR/rdfa-syntax/, Last visited: September 2012.

[34] CURIE syntax 1.0, http://www.w3.org/TR/curie/, Last visited: September 2012.

2.4.4 *MICRODATA*

With the similar motivation to RDFa in XHTML, HTML5 was created with a functionality to embed semantics in web pages. This functionality is called Microdata. Microdata introduces new attributes for semantic data. The full list of Microdata attributes can be found at the Microdata specification document[35]. A brief example of using Microdata is shown in the following code snippet.

```
<div itemscope itemid=”#publication01”>
  <p>Title: <span itemprop=”hasTitle”>What is
  Microdata?</span></p>
</div>
```

The example code contains a subject, a property and an object for a triple. Similar to the previous RDFa example, the subject is a fragment of the web page referred to as “publication01”. The property is “hasTitle”. The object is a string of text with the value “What is Microdata?” This triple can be serialised in a similar RDF/XML snippet as the example on RDFa.

2.5 SEMANTIC MAPPING

This section describes a data integration process called semantic mapping. The section begins with the definition of the term semantic mapping. After that, a framework for designing manual semantic mapping tool called the *cognitive support framework* is presented. Finally, an automated semantic mapping tool called the Alignment API is presented.

2.5.1 *DEFINITION OF SEMANTIC MAPPING*

As defined in section 2.1, a web data extraction process begins with identifying relevant data on the web. After the data is identified, it may not be in the same format or the same structure as the one used in the application that is extracting the data. Therefore, the identified data has to be mapped into a new structure. With the data of interest in this thesis being semantic markup data, the mapping technique called *semantic mapping* is found to be relevant and thus chosen.

[35] HTML Microdata, http://www.w3.org/TR/microdata/, Last visited: September 2012.

In this thesis, semantic mapping refers to a data integration process also known as ontology mapping. The terms ontology mapping, ontology matching and ontology alignment are used interchangeably in literature (Fu 2011). Ontology mapping is defined as “the task of relating the vocabulary of two ontologies that share the same domain of discourse in such a way that the structure of ontological signatures and their intended interpretations are respected” (Kalfoglou and Schorlemmer 2003). An ontology mapping process generates correspondences between two ontologies (Euzenat and Shvaiko 2007). These correspondences establish relationships between terms in two ontologies. Therefore, descriptions of a resource using one ontology can be translated to another. This allows instances to be duplicated from source to target ontology.

The term *semantic mapping* is used in this thesis instead of *ontology mapping* to avoid the implication of data complexity in the term *ontology*[36]. However, the same technique of identifying correspondences is used. Ontologies such as those published in the Web Ontology Language (OWL) format[37] can be much more expressive than knowledge models in the scope of this thesis. For example, an OWL ontology may have a restriction on the number of objects of a property. A knowledge model constructed from semantic markup data, on the other hand, is unlikely to be capable of having such restriction[38].

In this thesis, the generation of each mapping correspondence is called *matching*. Matching algorithms can be classified by the data they process (Shvaiko and Euzenat 2013). Terminological matching algorithms compare strings of text. Structural matching algorithms compare structures. Extensional matching algorithms compare instances. Semantic matching algorithms use logical reasoning to deduce correspondences (Shvaiko and Euzenat 2013).

Semantic mapping can be done manually as well. There has been research on reducing human effort in manual mapping through development in user interface. The following sub-section describes the cognitive support framework, a framework for manual mapping tools.

[36] Ontologies, http://www.w3.org/standards/semanticweb/ontology, Last visited: February 2013.

[37] Web Ontology Language, http://www.w3.org/2004/OWL/, Last visited: September 2012.

[38] Technically, semantic markup can be used to annotate HTML elements with OWL. However, in practice, the author of the thesis has never come across any web page that uses OWL’s features in semantic markup.

2.5.2 *MANUAL SEMANTIC MAPPING*

Fully automated semantic mapping is still imperfect and often requires supervision from human (Falconer et al. 2006). An approach to reducing human effort in semantic mapping is in tools used in semantic mapping tasks. COMA++ (Aumueller et al. 2005) was proposed as a graphical user interface for semantic mapping but emphasised little on the role of the human user. AgreementMaker (Cruz et al 2009) offers sophisticated mapping algorithms and a graphical user interface. Howev0065r, AgreementMaker does not support relationships amongst entities other than superclass–subclass.

To the best of the author's knowledge, the most relevant work on manual semantic mapping is CogZ (Falconer and Storey 2007). In CogZ, a set of use cases for the involvement of the human user in semantic mapping is proposed. CogZ is also implemented as an extension to the Protégé (Noy et al 2001) ontology editor. CogZ shows two knowledge models side by side and lets the user map them manually. In the design, CogZ proposes a set of use cases called the *cognitive support framework*. The use cases can be found as follows.

1. **Analysis and Decision Making**

   1.1. **Discover mappings**[39]
   Users should be able to create and explore temporary mappings.

   1.2. **Make mapping decisions**
   Users should be able to accept or reject a suggested mapping.

   1.3. **Inspect definition of term**
   Users should be able to have access to definitions of ontology[40] terms.

   1.4. **Inspect context of term**
   Users should be able to see how a term is used in an ontology.

[39] In the opinion of the author of the thesis, the term "mapping" used in this paper on CogZ is used in a more loose sense than in this thesis. "Mapping" in the paper means either identifying a relationship between terms in an ontology or identifying relationship between ontologies. In this thesis, the former is referred to as "matching" and the latter is referred to as "mapping".

[40] This thesis uses the term "knowledge model" or simply "model" in place of "ontology". This is because this thesis is focused on semantic markup which is less expressive than conventional ontology formats like OWL.

2. **Interaction**

   2.1. **Explore ontologies**

   Users should be able to navigate through terms in an ontology.

   2.2. **Explore and verify potential mappings**

   Users should be able to accept or reject potential mappings.

   2.3. **Explore and remove verified mappings**

   Users should be able to remove verified mappings.

   2.4. **Perform search and filter**

   Users should be able to search and filter terms in an ontology.

   2.5. **Direct creation and manipulation of the mappings**

   Users should be able to add metadata to verified mappings or create mappings manually.

3. **Analysis and Generation**

   3.1. **Generate mappings**

   There should be automatically generation of mappings to help users identify simple mappings.

   3.2. **Execute mappings**

   Users should be able to test mappings by transforming instances from source to target ontologies.

   3.3. **Save verification state**

   Users should be able to save a state of mapping in case of interruptions to the mapping session.

   3.4. **Conflict resolution and inconsistency detection**

   Users should be able to detect conflicts and inconsistencies in the mappings.

4. **Representation**

   4.1. **Source and target ontologies**

   The tool should provide a visual representation of ontologies.

4.2. **Potential mappings**

The tool should provide a representation of potential mapping correspondences with description and context.

4.3. **Verified mappings**

The tool should indicate whether a mapping correspondence is verified.

4.4. **Identify "candidate-heavy" regions**

The tool should identify and display parts of ontologies that contain a relatively large number of candidate mapping correspondences.

4.5. **Identify possible starting points**

The tool should automatically identify areas of ontologies where mapping correspondences are most likely to be verified.

4.6. **Progress feedback**

The tool should inform users of their progress in mapping.

4.7. **Reason for suggesting a mapping**

The tool should provide reasons behind a suggestion for a mapping correspondence.

The highly-detailed use cases of CogZ are useful for implementing a manual semantic mapping tool. Therefore, CogZ is chosen as the state-of-the-art work to build upon in this thesis.

2.5.3 AUTOMATED SEMANTIC MAPPING

From the previously mentioned cognitive support framework, the use case number 3.1 suggests that an automated semantic mapping process could reduce human effort. A recent survey (Shvaiko and Euzenat 2013) categorises automated semantic mapping tools by a number of features. To integrate an automated semantic mapping tool into a larger semantic mapping tool, the automated tool should provide a method to extend or include it in another tool. From the survey, the tools relevant to this thesis are Falcon (Hu et al. 2008) and the Alignment API (Euzenat 2004).

Falcon

Falcon offers a graphical user interface that shows the automatically generated mapping correspondences. Falcon provides both terminological and structural mapping algorithms.

Falcon is written in Java and provides Java classes for download. However, the author of the thesis found extending or integrating Falcon in to a new application to be non-trivial.

Alignment API

The Alignment API provides an application programming interface (API) for custom matchers. It also provides an API to be integrated in a Java program. With the Alignment API's ability of using arbitrary matching algorithms, its ease of integration, and its stable use in the research community, it is chosen as the automated semantic mapping tool used in this thesis.

2.6 Summary

In this chapter, background knowledge on web data extraction, web pages, linked data, semantic markup and semantic mapping was presented. Web data extraction is a research area founded after the World Wide Web became a widely used channel for data publication. A web data extraction process consists of data identification and mapping of the identified data to a new format or structure.

Web pages are constructed using markup languages, mainly HTML. Derivatives of HTML have been developed with the most recent being XHTML and HTML5. XHTML was developed to make web pages fully compatible with XML applications. XHTML inherited XML's strict parsing rules. HTML5 was created to be an alternative to XHTML as an updated version of HTML.

Linked data is a method of publishing structured data on the web. With linked data, entities in data documents can refer to other entities in other documents. This creates a global data space. Data published as linked data is published in the RDF format. A set of RDF data consists of statements, which are composed of a subject, a property and an object. These statements are also called triples.

Linked data can be embedded in web pages through semantic markup. Semantic markup is a method of annotating HTML elements with terms that would be composed into triples. Three current major semantic markup standards are Microformats, RDFa and Microdata.

Semantic mapping is a data integration method focused on knowledge models. In semantic mapping, correspondences are created between terms manually or by automated algorithms.

Mapping correspondences indicate relationships between those terms and can be used to transfer instances from one model to another. The manual semantic mapping tool in this thesis will be designed based on the cognitive support framework. The automated mapping tool will be based on the Alignment API.

3 STATE OF THE ART

This chapter discusses the on-going research related to the topic of this thesis and concludes with an assessment of subjects for further contributions.

3.1 INTRODUCTION

As discussed in section 2.4, semantic markup allows data on a web page to be identified through annotation of HTML elements. This creates an opportunity for using semantic markup in the data identification part of a web data extraction process. To build upon existing research, a survey was conducted by the author of the thesis on both semantic markup and web data extraction. The survey on semantic markup consists of applications for publishing and consuming semantic markup data. The survey on web data extraction explores what type of web data is being extracted and whether semantic markup data is used.

The rest of the chapter begins with a survey of the current applications of semantic markup. This survey shows how semantic markup is currently being published and consumed by applications. This chapter proceeds with a discussion of current approaches in web data extraction. The discussion shows how those techniques are applicable to the problem under study or how they could benefit from the use of semantic markup. After that, a survey of recent semantic mapping tools and mapping evaluation techniques is presented. The chapter then concludes with an assessment of the gaps for research contributions in these related areas.

3.2 APPLICATIONS OF SEMANTIC MARKUP

As a method of publishing data, applications of semantic markup can be classified into two groups: applications for publishing semantic markup data and applications for consuming semantic markup data. The first group largely consists of making publishing semantic markup data easier through user interface development. The first group of applications also includes methods for publishing semantic markup data in a specific area of interest. The second group of applications includes the way that the data is used by search engines and how it assists and personalises content display in web browsers. This second group also includes preliminary work on data extraction from web pages containing semantic markup data.

3.2.1 PUBLISHING SEMANTIC MARKUP DATA

This sub-section begins with a comparison of the approaches to publishing semantic data of the three major semantic markup specifications: Microformats, RDFa and Microdata.

Unlike the rest of the three major semantic markup specifications, Microformats does not allow references to external vocabularies. Microformat publishers need to know the terms defined by the Microformats community in addition to the HTML specifications.

RDFa and Microdata allow the use of arbitrary vocabularies. Publishers have to know which vocabulary to use and how to embed the data in HTML source code. Tools with graphical user interfaces have been developed to assist the publication process.

A recent survey collected and compared user interfaces of tools for publishing knowledge models, or what the authors called “semantic authoring” tools (Khalili and Auer 2012). The survey classifies approaches to semantic authoring tools into two groups: *top-down* and *bottom-up*. The top-down approaches aim at helping to create semantic content in a formal ontology format. The bottom-up approaches aim at adding semantic data into existing documents. Annotation of HTML elements with semantic markup data belongs to the bottom-up group.

Semantic markup data publication tools that the survey chose to analyse are Loomp (Luczak-Rösch and Heese 2009) and RDFaCE (Khalili et al. 2012).

Loomp is aimed towards ordinary web users especially journalists, who are not necessarily familiar with linked data. It proposes a one click annotation user interface. The interface allows the user to embed RDFa data into a text HTML element in a similar fashion as applying text formatting in word processors.

RDFaCE is an adaptation of a what-you-see-is-what-you-get (WYSIWYG) text editor. It allows the user to annotate a piece of text with RDFa data as well as discover new vocabularies.

RDFauthor (Tramp et al. 2010) is another recent application for publishing RDFa data. It is aimed to web pages that already have RDFa data. RDFauthor displays the data in *widgets* – a user interface component similar to dialogue boxes – which let the user edit the data. These widgets have different data fields depending on the recognised namespace from the RDFa data. The data is then submitted as an update query.

Most of the recent research on semantic markup data uses RDFa as the specification of choice for semantic markup. However, it is very likely that these applications can be adapted to use other specifications as well. This is because all three major semantic markup specifications are based on the concept of RDF triples.

The next sub-section presents applications for consuming semantic markup data.

3.2.2 Consuming Semantic Markup Data

Annotating HTML elements with semantic markup data makes parts of the data on the web page immediately identifiable by a program. Publishing metadata via semantic markup data is becoming more and more widespread (Mühleisen and Bizer 2012). A variety of applications have been built on identifying data through semantic markup. The author of the thesis categorises these applications into search engines, client-side display augmentation, personalisation and web data extraction.

Search Engines

Search engines are one of the key actors that consume semantic markup data. Google has a set of metadata that, if used to annotate HTML elements on a web page, will be displayed in the search results. This set of data is called *rich snippets* (Steiner et al. 2010). Rich snippets include Microformats, RDFa and Microdata[41]. Rich snippets not only benefit content publishers in exposing their products, rich snippets also help Google's crawler ensure that the retrieved data is relevant. For example, instead of having to recognise different date and time formats in plain text, the crawler can look for HTML elements annotated with semantic markup data that describes date and time.

Rich snippets are an example of how data on a web page can be identified through semantic markup and how the data can be used.

[41] Rich snippets (microdata, microformats, RDFa, and Data Highlighter), http://support.google.com/webmasters/bin/answer.py?hl=en&answer=99170, Last visited: March 2013.

Client-side Display Augmentation

Domain-specific content can be identified and recognised if it contains semantic markup data. Such content may benefit from an alternative interface – augmenting normal web page rendering. For example, a web browser can be programmed to detect a textual chemical formula from its accompanying semantic markup data. The browser then can use the data to query the formula and display a diagram of the molecule instead (Willighagen and Wikberg 2010).

Personalisation and Localisation

Semantic markup data is published on web pages which are downloaded to the client side. Applications on the client side can then use semantic markup data to personalise data such as currency or date and time by location (Al-Jabari et al. 2009). Having HTML elements annotated also enables tracking of user behaviours such as identifying which element is clicked on (Plumbaum et al 2012). The user behaviour data can then be analysed and content that is potentially more interesting to the user can be delivered.

Web Data Extraction

There has not been much research on using semantic markup data in web scraping or web data extraction[42]. In the author's opinion, the consumption methods mentioned above use semantic markup data as a temporary vehicle of RDF graphs. There is no emphasis on merging and storing the extracted data into one central model.

A recent paper associates semantic markup data to the discipline of web data extraction (Geel et al. 2012). It proposes a framework for extracting different formats of structured data and merging the data into one central model. The supported formats are Microformats, RDFa and Microdata. This poses semantic markup as a new resource for web data extraction. However, the process of converting semantic markup data to a common format like RDF is trivial. This means there is an opportunity to investigate if a web data extraction process could benefit more from semantic markup data. For example, the association between semantic markup data and the annotated HTML elements can be exploited to identify data outside the data defined by semantic markup.

[42] For example, a search on the book series The Semantic Web: Research and Applications from 2009 – 2012 with the keyword "RDFa" yields 20 papers. None of the papers is focused on using RDFa in web data extraction.

A survey on the state of the art research on web data extraction is discussed in the next section. The survey is aimed to find the area that would benefit from semantic markup.

3.3 Semantic Markup Data and Web Data Extraction

As mentioned in the motivation of this thesis in section 1.1, there has been little research in using semantic markup in web data extraction. Web data extraction techniques have been developed as the web changes. With semantic markup as a new resource of web data, a survey was conducted to see whether it has been adopted by the research area. Recent surveys show a diversity of tools and techniques. However, there are still challenges. Addressing some of the challenges might benefit from exploiting semantic markup data.

3.3.1 Input for Web Data Extraction Systems

To verify the novelty of using semantic markup data in web data extraction, a survey was conducted on web data extraction systems. A paper by Chang et al. categorises variations of input to web data extraction systems (Chang et al. 2006). According to the paper, input documents to web data extraction systems can be categorised as follows:

- **Semi-structured documents:** These documents contain data that is machine-readable. However, they also contain data that is published for human to read and not immediately machine-readable. Web pages belong to this category.

- **Free-text documents:** These documents are published for human consumption and are not immediately machine-readable.

- **Templates:** Templates are a pre-defined structure and patterns of data.

The paper also categorises web data extraction systems based on how they are operated. According to the paper, web data extraction systems and their input can be categorised as the following list.

- **Manually constructed systems:** In this category, wrappers are manually created specifically for each website. This method requires the user to be proficient in programming. Web data extraction systems in this category use semi-structured documents and templates as inputs.

- **Supervised systems:** In this category, the user creates an example set of labels for data on a web page. The system then uses the example labels to extract data from similar web pages. The user may use these systems through a graphical user interface and does not have to be proficient in programming. Web data extraction systems in this category use free-text or semi-structured documents as inputs.

- **Semi-supervised systems:** In this category, web data extraction systems receive an example set of labels similar to that of supervised systems. However, the example labels do not need to be complete and exact. The systems in this category generate extraction rules from examples. Web data extraction systems in this category use templates.

- **Unsupervised systems:** Web data extraction systems in this category do not need any example labels or user interaction. Web data extraction systems in this category also use templates.

Another survey paper focuses on semi-supervised and unsupervised web data extraction systems (Fiumara 2007). This paper also considers web pages as semi-structured documents. The paper includes the Dynamo project (Bossa et al. 2006), a web data extraction systems that uses "semantic structures" in a form of metadata in <META> tags to locate information of interest. Although this is not semantic markup as defined in this thesis, the author of the thesis believes it shows that web data extraction may benefit from document annotation.

A recent survey paper lists techniques for extracting web data (Ferrara et al. 2012). These techniques can be categorised into two groups: web wrappers and machine learning. The paper also describes the input of each technique.

- **Web Wrappers:** As discussed in section 2.1, wrappers are programs that identify data in web content. The paper lists three approaches of web wrappers: regular expression-based approach, logic-based approach and tree-based approach.
    - **Regular expression-based approach:** A wrapper using this approach uses a set of manually created text extraction rules called regular expression. The wrapper would apply the rules to a web page's source code and detect areas of data such as word boundaries, tags or table structures.

  - **Logic-based approach:** A wrapper using this approach is created as a program. The wrapper would look at a web page as both its semi-structured nature as a document object model[43] and its textual nature. The paper suggests that because the wrapper of this kind is created programmatically, it could be "incorporated into visual tools". Therefore, the wrapper could allow the user to "[highlight] elements of the document and [specify] relationships among them". The wrapper could also allow the user to apply arbitrary conditions on data extraction.

  - **Tree-based approach:** A wrapper using this approach uses the DOM representation of a web page. By looking at a web page as a tree data structure, existing information extraction algorithms that work on tree data structures can be used on a web page as well. An example of this approach is the DIADEM project (Furche et al. 2012) which aims for data extraction from interactive websites. However, there is no mention of semantic markup in the DIADEM project.

- **Machine Learning:** According to the survey, web data extraction systems in this group are aimed towards domain-specific data extraction. With this technique, web data extraction systems learn from a set of manually labelled web pages and create extraction patterns to use on unseen web pages. There are also hybrid systems that balance between automatic extraction and human engagement.

The concept of semantic markup is almost never mentioned in all of the selected surveys. The author of the thesis thinks that; in this paper's categorisation system, web pages that contain semantic markup data would be considered as semi-structured. This is because there is much diversity of web content. It is unlikely that web content publishers could find vocabularies that fit for annotating every part of the content. However, semi-supervised and unsupervised web data extraction systems may be able to exploit semantic markup data. Thus these systems could be used on semi-structured input documents or web pages. This will be discussed later on in sub-section 3.3.3.

[43] See section 2.2 for the description of the document object model.

The latest survey (Ferrara et al. 2012) also shows that links between visual representation of a web page and the underlying data can be used in web data extraction. However this use of the links between visual data and underlying data does not mention the use of semantic markup. The paper does mention Microformats briefly. However, it was to show Microformats as another web source apart from web pages. The author of the thesis thinks that Microformats in fact has close relationships to the web page. Therefore it should not be viewed separately from the web page.

This thesis also presents challenges in current web data extraction systems. These challenges are discussed in the following sub-section.

3.3.2 Challenges

Surveying the state of the art research on web data extraction showed that there are still challenges in the field. The latest survey up to the point of writing this thesis summarises the current challenges in the following list (Ferrara et al. 2012).

- **Help from human experts:** A high level of automation in web data extraction is desirable. However, in some cases, a minimal amount of human efforts may significantly increase accuracy of extraction. The challenge lies in balancing the trade-off between the levels of automation and human efforts. This challenge is also addressed in a previous survey (Chang et al. 2006). The previous survey concludes that “unsupervised approaches can only support template pages” and “supervised approaches, although require annotations from users, extend well to non-template page extraction if proper features are selected for extraction rules”.

- **Large volumes of data:** Certain applications of web data extraction such as business intelligence have to process a large amount of data on a timely basis.

- **Privacy:** Certain applications of web data extraction such as social network data extraction have to provide privacy guarantees.

- **Large training sets for machine learning:** Machine learning-based approaches to web data extraction still require large training sets which are expensive to create in terms of time and human efforts.

- **Change in data structure:** Web data is prone to change over time or across different websites. To cope with changes, web data extraction systems should have a level of independence from a certain data structure.

Semantic markup may contribute to addressing some of these challenges. This is discussed in the following sub-section.

3.3.3 Possible Benefit from Semantic Markup

By surveying recent research on web data extraction, it appears that semantic markup data is not used much, if at all, in the field. Introducing semantic markup data as a new type of data in web pages may help address the challenges perceived in the area. The author of the thesis sees opportunities in the following challenges:

- **Reducing human efforts:** Semantic markup data is published by humans as a means to convey semantics in web content in a machine readable way. The author considers that the data published this way requires less interpretation from humans. The process of identifying relevant data in web data extraction systems may begin from semantic markup data instead of the less semantic HTML page.

- **Uniform data structure:** Current semantic markup standards can be converted to RDF[44][45][46]. With RDF being a dominant data format of the semantic web (Bizer et al. 2009); it is likely that any new semantic markup standards would be convertible to RDF as well. If web content publishers who use semantic markup adhere to the specifications, semantics in the content will be easy to extract while web page structures can vary.

Semantic markup is still a new resource in web data extraction. The use of semantic markup may be a solution to some of the challenges faced in the field of web data extraction.

3.4 Conclusion

[44] “Microformat FAQs relating to RDF”, http://microformats.org/wiki/faqs-for-rdf, Last visited: March 2013.

[45] W3C, “RDFa 1.1.Primer”, http://www.w3.org/TR/xhtml-rdfa-primer/, Last visited: March 2013.

[46] W3C, “Microdata to RDF”, http://www.w3.org/TR/microdata-rdf/, Last visited: March 2013.

From the review on research in the related areas, the author finds that semantic markup data is a relatively new resource in web data extraction. Challenges in the area of web data extraction have been documented. The author believes that some challenges in web data extraction could be addressed by introducing semantic markup data.

4 WEB DATA EXTRACTION HIGH LEVEL USE CASE

In this chapter, the web data extraction problem focused on in this thesis is illustrated. It forms a generic use case for the rest of this dissertation and illustrates limits to the scope of this work.

4.1 HIGH LEVEL USE CASE

The problem attached in this thesis is data extraction from web pages describing a single concept instance or single-instance web pages. An example of such a web page is a web page that describes a product or person rather than a collection of things. This means the URL of the web page is appropriate to use as the URI of an instance of a class. For example, the Internet Movie Database entry for the film *An Education*[47] has the URL "http://www.imdb.com/title/tt1174732/". This URL is appropriate as a URI of an instance of a class describing films because the URL is unique and points to a web page describing a film.

There are websites that collect or aggregate data from single-instance web pages. The service provided by these websites can be referred to as a data aggregation service. This kind of website exists in many domains such as retail products[48], films[49] and news[50]. The data aggregation website typically extracts data from other websites in the domain and displays the data together in one place. The data from different sources are mapped to an internal database schema or knowledge model. In this way the data can be categorised, queried and displayed in a uniform fashion. This allows the user to browse for information without having to visit multiple websites.

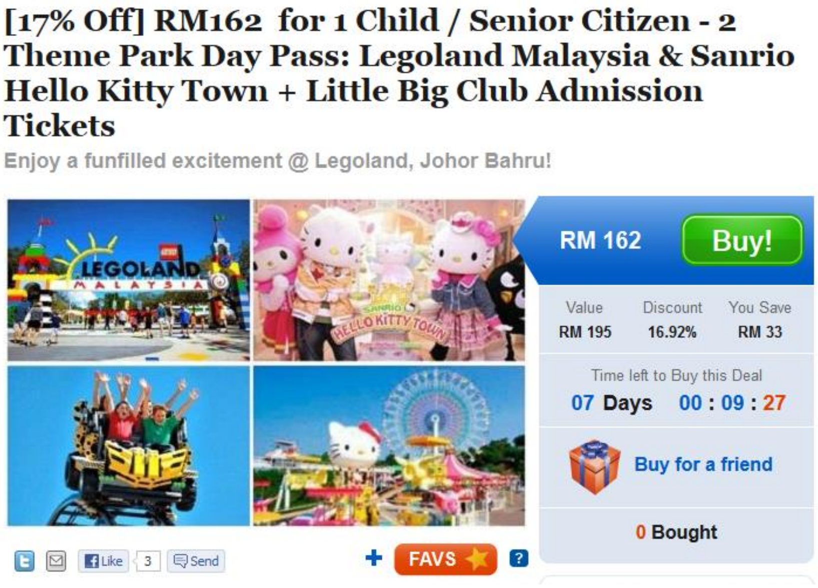
[47] An Education is a 2009 drama film directed by Lone Scherfig.

[48] For example, Pricepinx (http://pricepinx.com) monitors product prices from 30 online retailers.

[49] For example, the Internet Movie Database (http://imdb.com) provides film show times from local cinemas.

[50] For example, the Feedly (http://www.feedly.com) displays news content from different source in a uniform interface.

To illustrate an example situation, Figure 1, Figure 2 and Figure 3 show three screenshots taken from three discount coupon websites: StarDeal.my[51], Deal4Real.asia[52] and Dealshelve[53], respectively.



[17% Off] RM162 for 1 Child / Senior Citizen - 2 Theme Park Day Pass: Legoland Malaysia & Sanrio Hello Kitty Town + Little Big Club Admission Tickets

Enjoy a funfilled excitement @ Legoland, Johor Bahru!

RM 162

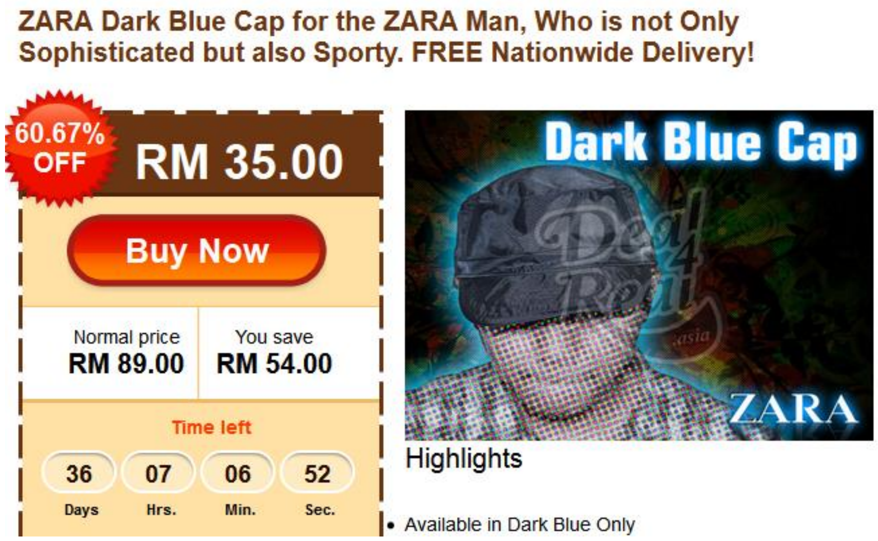
Value	Discount	You Save
RM 195	16.92%	RM 33

Time left to Buy this Deal

07 Days 00 : 09 : 27

0 Bought

Figure 1: Example Discount Coupon from StarDeal.my



ZARA Dark Blue Cap for the ZARA Man, Who is not Only Sophisticated but also Sporty. FREE Nationwide Delivery!

60.67% OFF

RM 35.00

Normal price	You save
RM 89.00	RM 54.00

Time left

36	07	06	52
Days	Hrs.	Min.	Sec.

Highlights

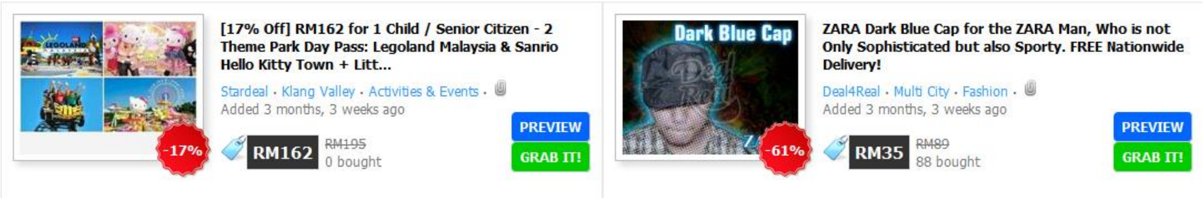
- Available in Dark Blue Only

Figure 2: Example Discount Coupon from Deal4Real.asia

[51] StarDeal.my, http://www.stardeal.my, Last visited: April 2013.

[52] Deal4Real.asia, http://deal4real.asia, Last visited: April 2013.

[53] Dealshelve, http://dealshelve.com, Last visited: April 2013.



[17% Off] RM162 for 1 Child / Senior Citizen - 2 Theme Park Day Pass: Legoland Malaysia & Sanrio Hello Kitty Town + Litt...

Stardeal · Klang Valley · Activities & Events ·

Added 3 months, 3 weeks ago

ZARA Dark Blue Cap for the ZARA Man, Who is not Only Sophisticated but also Sporty. FREE Nationwide Delivery!

Deal4Real · Multi City · Fashion ·

Added 3 months, 3 weeks ago

Figure 3: Aggregated Coupons on Dealshelve

It can be seen from the figures that data such as coupon titles, prices and images from Stardeal.my and Deal4Real.asia are extracted and shown in the same structure on Dealshelve. It has been shown[54] by the developers of Dealshelve that the website uses a web data extraction software library called Scrapy[55]. The software library is used by Dealshelve to extract data from other websites that do not provide an application programming interface (API) for data aggregation.

4.2 DATA EXTRACTION FROM SINGLE-INSTANCE WEB PAGES

The web data extraction process, consisting of identifying the data of interest and mapping the data to other data models, has been illustrated in previous work (Embley et al. 1999, Chang et al. 2006). This web data extraction process, as focused on in this thesis, is illustrated in Figure 4.

From the illustration, the process is divided into two major steps: data identification and mapping. In data identification, data from the target web page is identified by a data identification tool set. The data identification tool set contains a set of algorithms for automated data identification. The data identification tool set also allows the user to manually identify the data through a user interface. The identified data is serialised into a data model defined by the source web page. In this thesis, this data model is called the *external model*.

[54] There is a conversation on the mailing list of the software framework Django regarding how DealShelve extract data from other websites. https://groups.google.com/forum/#!topic/django-users/tatrT28o-pU, Last visited: April 2013.

[55] Scrapy, http://scrapy.org/, Last visited: April 2013.

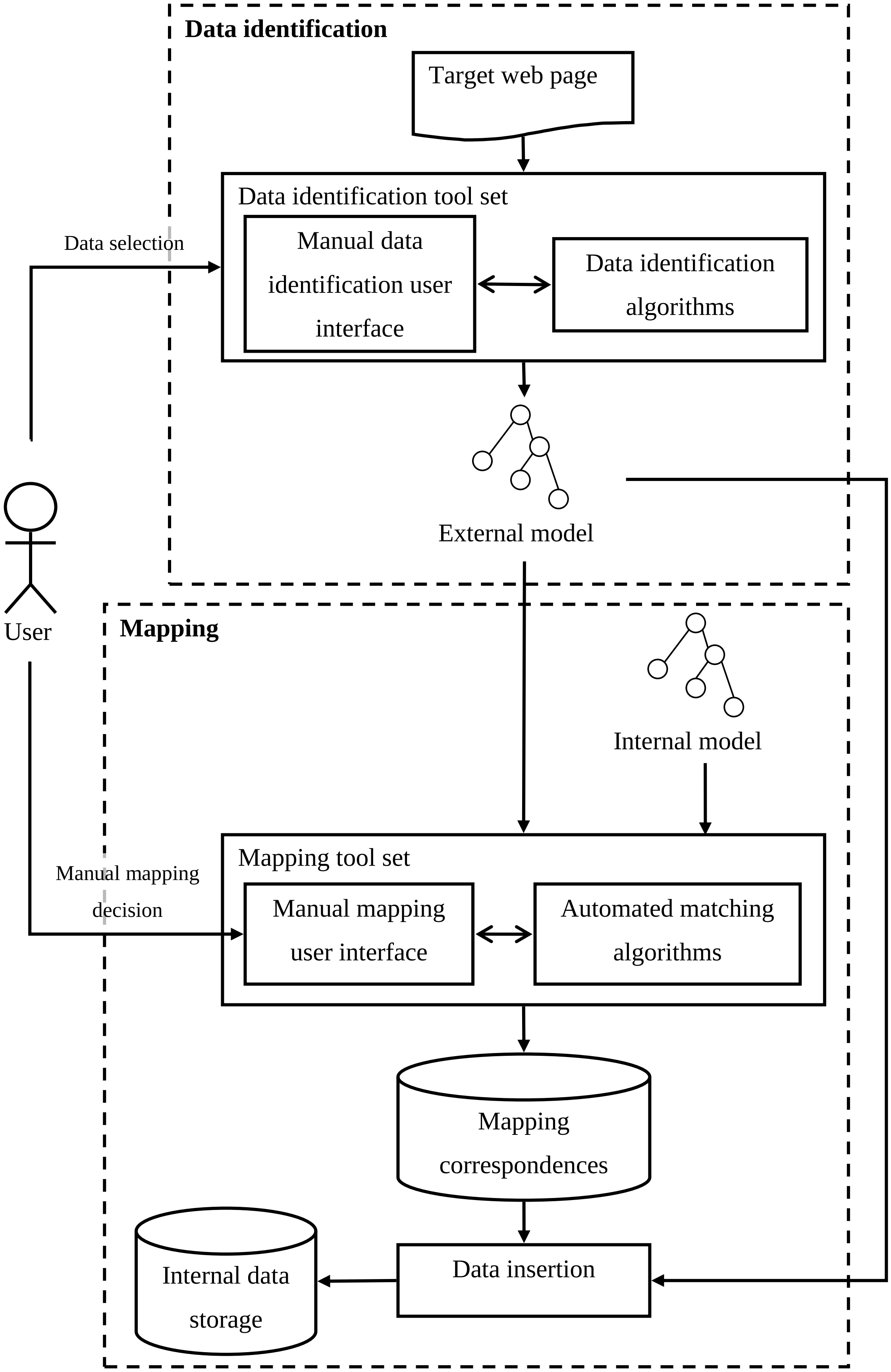


Figure 4: Web Data Extraction Process for Single-instance Web Pages

After the external model data is identified, it is necessary to insert data from the web page into the internal data structure. To support this insertion process, relationships between data in the external model and data in the internal data storage must be defined. These relationships are referred to as *mapping correspondences*. The known structure of the data in the internal data storage is represented by a data model called the *internal model* in this thesis. All mapping correspondences are between the external model and the internal model.

To generate mapping correspondences, both the external model and the internal model are passed into a mapping tool set. The mapping tool set, in a similar fashion to the data identification tool set, allows both manual and automated mapping. After mapping correspondences are generated, they are used to perform data insertion on the external model. This places the external data into the internal storage.

After this combined process has been executed for a single target web-page it may start again with another target, but this is outside the scope of the current work.

To further illustrate the web data extraction process for single instance web pages, following example mapping correspondences are created from example web pages shown in section 4.1. From the examples, the website Dealshelve aggregates discount coupon data from other websites, StarDeal.my and Deal4Real.asia, through the use of a web data extraction software library called Scrapy and possibly with human editing. The aggregated data from both websites, along with data from other websites, is then displayed in a uniform manner. This shows that the data from two websites with likely different data structures was stored in a uniform data model on Dealshelve.

In this case, data from the websites StarDeal.my and Deal4Real.asia is the *external model* for DealShelve. Meanwhile, the data model in DealShelve that holds data from both websites is the *internal model*.

5 VISUAL CONTEXT IN SEMANTIC MARKUP MAPPING: DESIGN AND IMPLEMENTATION

This chapter presents the design and implementation of a semantic markup mapping tool. The aim of this semantic markup mapping tool is to be a part of and to improve the web data extraction process discussed in chapter 4. This chapter begins with section 5.1 explaining the motivation behind the design and implementation. Section 5.2 describes the design and Section 5.3 then proceeds to describe the implementation. Finally, section 5.4 summarises the chapter.

5.1 MOTIVATION

As discussed in section 2.1, the process of web data extraction consists of identifying the data of interest and mapping the structure of that data to a structure of a target data model. It was argued in section 2.4 that HTML elements on a web page could be annotated with semantic data by a form of metadata called semantic markup. In this way the author of the thesis believes the use of semantic markup is a way to identify data on a web page that can be extracted. An example of such use of semantic markup for data identification is the way search engines use semantic markup to add more data to the search results[56].

Semantic markup data can be serialised into an RDF graph[57]. This allows the semantic markup data to be mapped semantically to knowledge models that are also RDF graphs. The author of the thesis sees this as a web data extraction process based on semantic markup.

As discussed in section 2.4, fully-automated semantic matching algorithms are considered impractical and many semantic mapping tools for human use have been developed. Using semantic mapping tools require expertise in the subject domain and there has been research on improving the usability of the tools to make them require less expertise.

[56] See sub-section 3.2.2.

[57] See section 2.4.

Many of the existing semantic mapping tools can be operated on RDF files and therefore can be operated on RDF serialisations of semantic markup. However, semantic markup has a distinctive feature unlike other forms of knowledge model: it is used to annotate HTML elements and therefore has links to the annotated elements. The users performing mapping tasks on semantic markup would not only have the definitions of the terms in the model but could also see how those terms are used in semantic annotation. This extra visual information may help reduce the level of expertise required in using semantic mapping tools.

To the best of the author's knowledge, there have not been any semantic mapping tools developed especially for semantic markup embedded in web pages. A semantic mapping tool that exploits the visual information related to semantic markup data may help reduce the level of expertise required in using such mapping tools.

The next section describes the design of a semantic markup mapping tool that exploits visual information.

5.2 DESIGN

This section begins with sub-section 5.2.1 outlining the use cases of the semantic markup mapping tool. Sub-section 5.2.2 then lists the requirements for a software implementation.

5.2.1 USER ACTIVITIES

As discussed in sections 2.4 and 5.1, in comparison to other RDF-based knowledge models, a distinctive feature of semantic markup is that it has links to HTML elements. This feature should be exploited in the semantic markup mapping tool.

To design the semantic markup mapping tool, a set of user activities of a state-of-the-art semantic mapping tool has to be derived. New user activities involving links between semantic markup and HTML elements are then added to the state-of-the-art set.

At this stage, semantic mapping tools are primarily focused on manual mapping. To the best knowledge of the author of the thesis, CogZ (Falconer and Storey 2007) is the most relevant work on user-focused tool for manual mapping. A set of use cases in mapping called a cognitive support framework is introduced in CogZ. The framework categorises the use cases into four dimensions as discussed in sub-section 2.5.2.

From the cognitive support framework, the author of the thesis derived a set of use cases that the initial design of semantic markup mapping tool should have. Figure 5 illustrates those use cases. The source each use is identified by its number.

From Figure 5, the user has four types of interactions with the tool: **navigate model**, **manipulate candidate mapping correspondences**, **generate mapping correspondences** and **export mapping correspondences**. The total number of use cases is 12.

In the **navigate model** use case, the user can locate and view an internal model and an external model. The external model is generated by locating a web page that contains semantic markup data and by extracting the semantic markup data. The links between semantic markup data in the external model and the source HTML elements are shown to the user. The user is also able to search through both models for specific terms.

In the **manipulate candidate mapping correspondences** use case, the user can view and manipulate (approve and reject) candidate mapping correspondence. These candidates are generated automatically by a matching algorithm.

In the **generate mapping correspondences** use case, the user can create mapping correspondences manually in addition to automatically generated candidates.

Finally, in the **export mapping correspondences** use case, the user can save the mapping result into a local file.

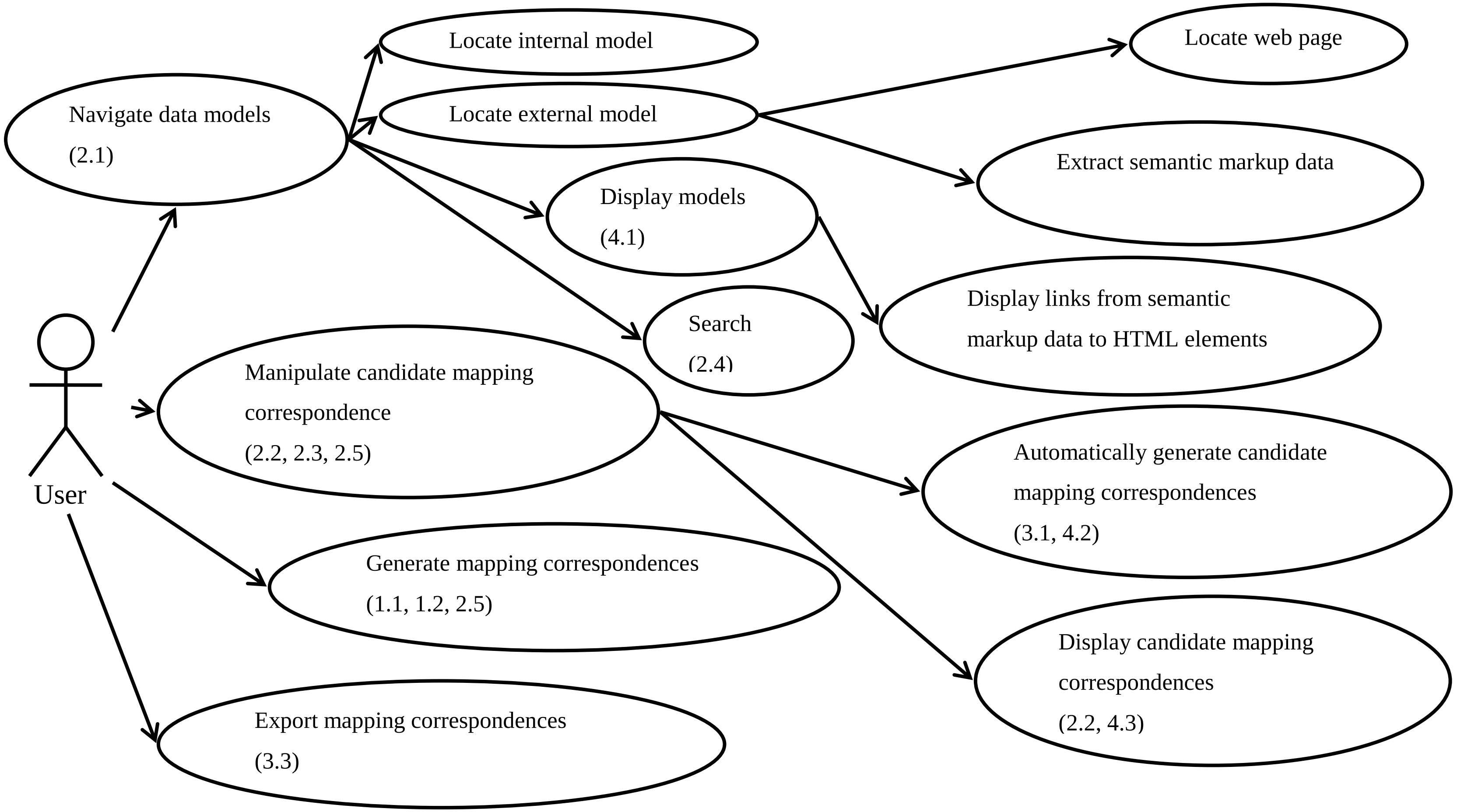

Locate internal model
Locate web page
Navigate data models
(2.1)
Locate external model
Extract semantic markup data
Display models
(4.1)
Display links from semantic
markup data to HTML elements
Search
(2.4)
Manipulate candidate mapping
correspondence
(2.2, 2.3, 2.5)
User
Automatically generate candidate
mapping correspondences
(3.1, 4.2)
Generate mapping correspondences
(1.1, 1.2, 2.5)
Display candidate mapping
correspondences
(2.2, 4.3)
Export mapping correspondences
(3.3)

Figure 5: Use Cases for Semantic Markup Mapping Tool

There are 9 use cases from the cognitive support framework that are not included in the above set of use cases. This is due to time limitation. The excluded use cases can be added in future implementations of the tool. The list of the excluded use cases and the reasons behind their exclusion are described below:

- **Analysis and Decision Making**

  In this dimension, use cases number 1.3, "inspect definition of term", and 1.4, "inspect context of term", are not included because the software implementation is undertaken as a web browser extension. For internal models, the user can look up the definition of the terms in the web browser. For external models, the terms are linked to how they are used on the web page. The user should easily see how the terms mean by looking at the annotated HTML elements.

- **Analysis and Generation**

  In this dimension, use cases number 3.2, "execute mappings", 3.3, "save verification state", and 3.4, "Conflict resolution and inconsistency detection", are not included.

  Use case number 3.2 is not included because of time limitation. The author of the thesis also sees that evaluation of a web data extraction process can be done at mapping correspondences without implementing mapping execution.

  Use case number 3.3 is not included because, from an observation by the author of the thesis, the number of terms in an external model is relatively low comparing to conventional knowledge models. Therefore, a mapping session of one web page is unlikely to have any interruptions or can be started over easily.

  Use case number 3.4 is not included because of time limitation.

- **Representation**

  In this dimension, use cases number 4.4, "identify 'candidate-heavy' regions", 4.5, "identify possible starting points", 4.6, "progress feedback" and 4.7, "reason for suggesting a mapping", are not included.

Use case number 4.4 and 4.5 are not included because external models have a relatively low number of terms. Therefore identifying a part of the model with a relatively large number of candidates mapping correspondence is considered not cost-effective.

Use case number 4.6 is not included because of the small number of terms in external models. The author of the thesis speculates that the user would be aware of their progress naturally as they navigate through the data model.

Use case number 4.7 is not included because, at the current stage, the proposed semantic markup mapping tool uses only one matching algorithm. Therefore, all the automatically generated candidates are suggested for the same reason.

This sub-section has identified the use cases of the semantic markup mapping tool. The next sub-section outlines the requirements for the software implementation.

5.2.2 REQUIREMENTS

The sub-section presents a list of requirements for a software implementation of the semantic markup mapping tool. These requirements are derived from the use cases described in the previous subsection. The list below outlines the requirements of the four main use cases: **navigate model**, **manipulate candidate mapping correspondences**, **generate mapping correspondences** and **export mapping correspondences**.

- **Navigate model**
    - The software implementation should be able to identify semantic markup data on a web page and serialise it into an RDF graph.
    - The software implementation should be able to identify and visualise the links between semantic markup data and its source HTML elements.
    - The software implementation should be able to read and visualise RDF graphs.
    - The software implementation should allow the user to perform a text-based search on the internal and external models.

- **Manipulation candidate mapping correspondence**
    - The software implementation should be able to use a matching algorithm to generate a set of candidate mapping correspondences.
    - The software implementation should have storage for an editable list of candidate mapping correspondences.
- **Generate mapping correspondence**
    - The software implementation should have storage for manually created list of mapping correspondences.
- **Export mapping result**
    - The software implementation should be able to serialise the mapping correspondence list into a file.

From the list of requirements for the software implementation, a user interface is designed. The next sub-section describes the design of a user interface for the semantic markup mapping tool.

5.2.3 USER INTERFACE

This sub-section presents the design of the user interface for the semantic markup mapping tool. From the requirements listed in the previous sub-section, the user interface is designed to have three components: **model display**, **visual clues** and **mapping correspondence display**.

Model Display

Semantic data graphs are usually visualised as at least one group of basic shapes (representing classes and individuals) linked by lines (representing properties). In a recent survey (Dadzie and Rowe 2011), 5 out of 8 selected linked data browsers provide such visualisation[58]. An example of this type of visualisation is RDF Gravity (Goyal and Westenthaler 2004) shown in Figure 6.

[58] It is referred to in the survey as a ‘graph view’.

With this research’s focus on web pages that represent single instances, it might be possible to simplify the display of semantic markup data on a web page as a tree instead of a graph[59]. In fact, Protégé (Noy et al. 2001), one of the state-of-the-art semantic data authoring tools, provides a tree display as the main interface for ontology manipulation. The tree display allows the user to see relationships among the data in a hierarchical fashion. Classes that act as subjects lead to their properties which then lead to their objects. The tree display also allows the user to temporarily hide parts they are not working on to make an efficient use of display space. Protégé also provides a graph display with less editing functionalities, i.e., the user cannot make changes to concepts. Figure 7 shows examples of these two Protégé interfaces.

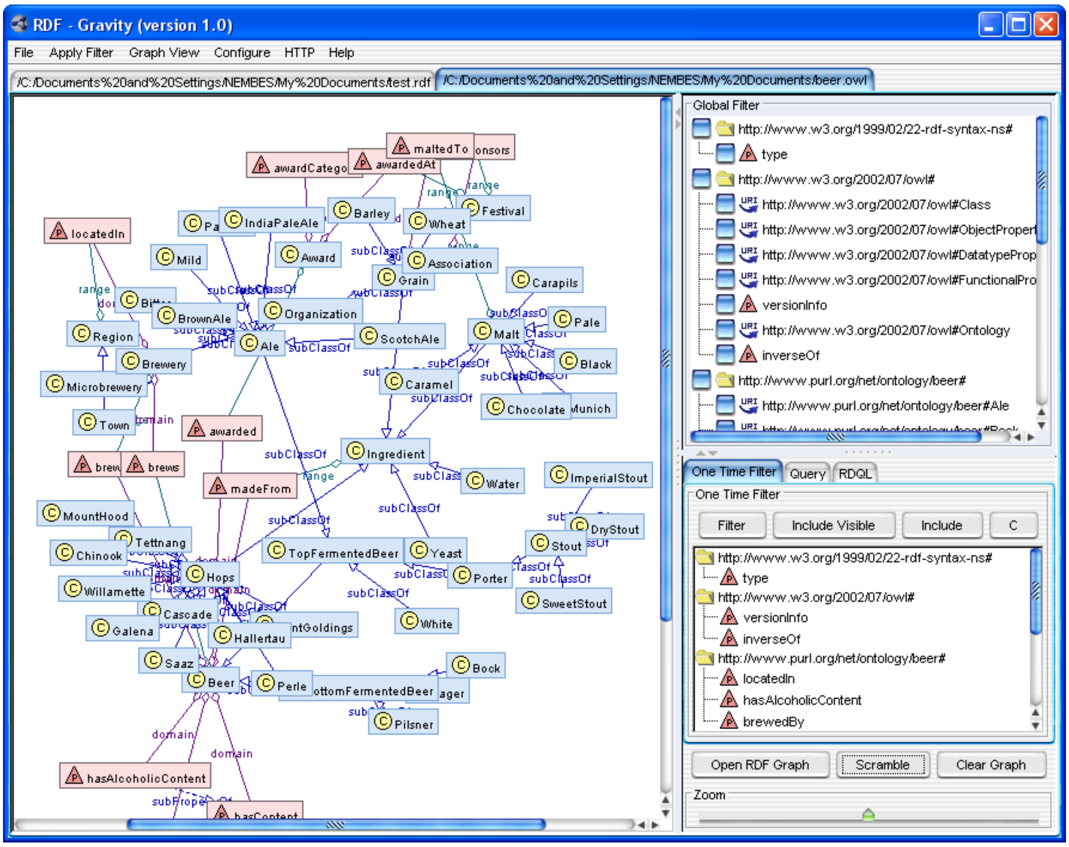


Figure 6: RDF Graph Visualisation in RDF Gravity, a Linked Data Browser

[59] In graph theory, a tree is indeed simply a connected graph without cycles. However, the term “tree” here is used to refer to a common user interface in software that displays hierarchy. One example of such interface is Windows Explorer’s collapsible folder structure.

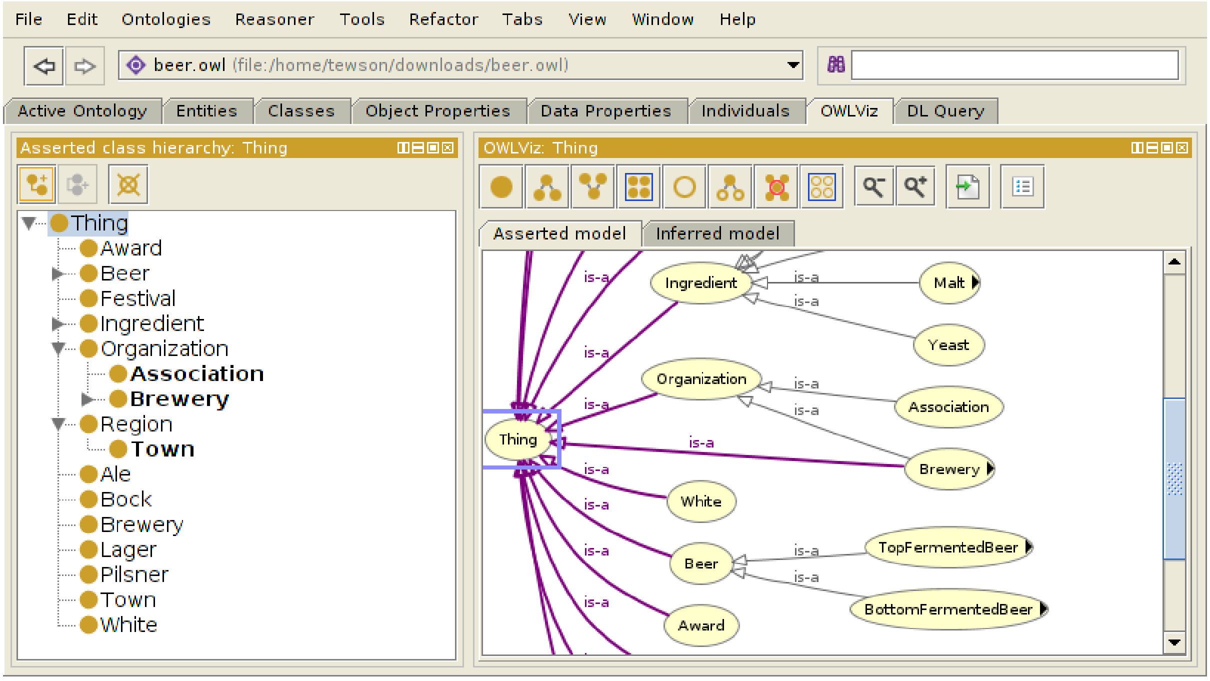


Figure 7: Class Hierarchy Tree and Graph in Protégé

The author of the thesis sees the tree display as a more practical approach to displaying knowledge models. Therefore, the tree display is used in the design of the semantic markup mapping tool.

Visual Clues

Links between semantic markup data and HTML elements can be found in the document object model representation of a web page. These links are defined where HTML elements are annotated with the markup. Since the understanding of DOM representation of web pages is beyond the scope of the intended mapping process, the user should not have to navigate the DOM itself to discover these links.

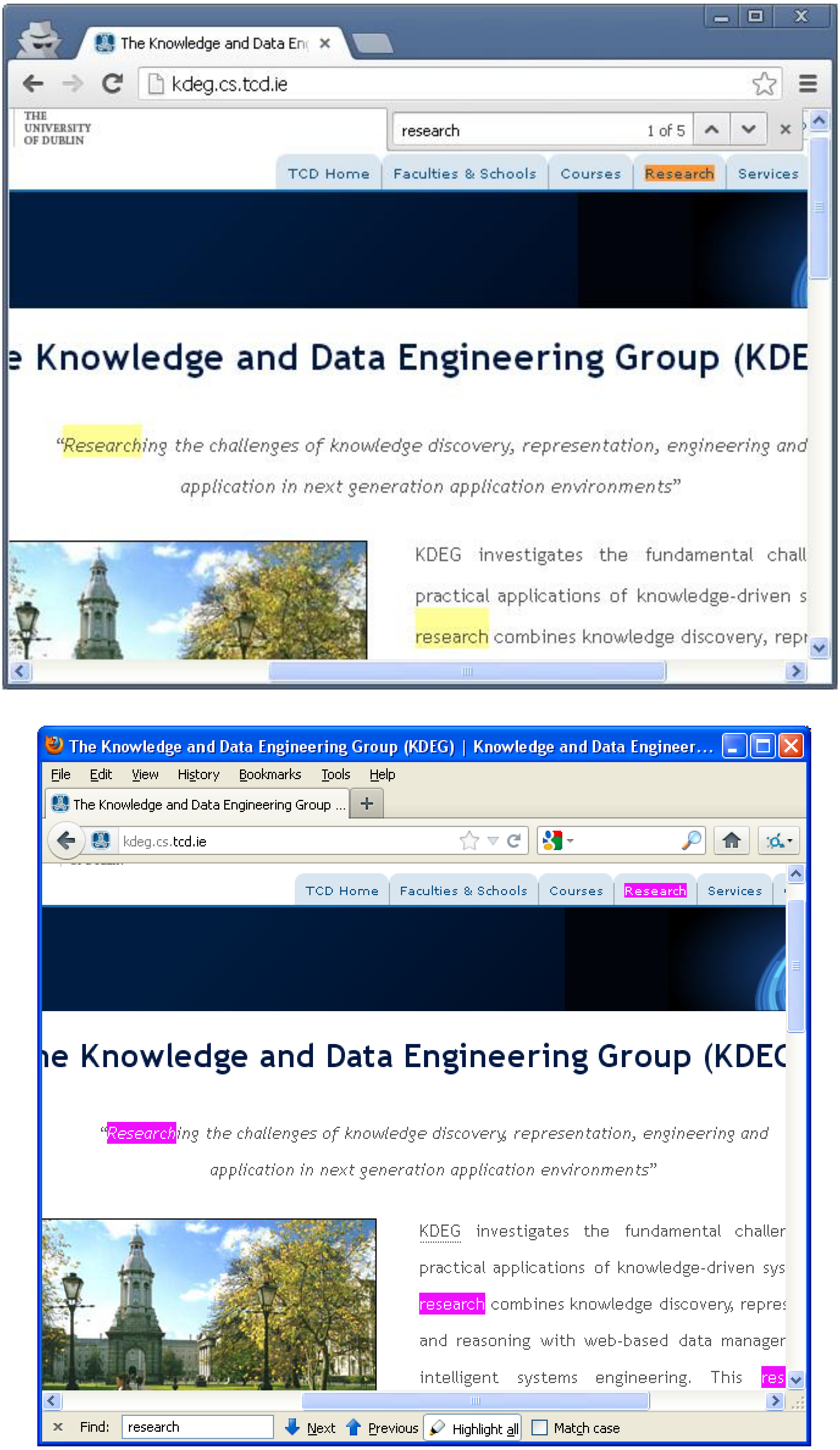
The task of exposing the links between semantic markup data and HTML elements can be broken down to two steps. The first step is to **locate and assign visual clues to HTML elements** that are annotated with semantic markup data. The next step is to **visualise the links** between the semantic markup data (displayed in the model display component described above) and the HTML elements.

In the tool design, locating annotated HTML elements is viewed as a similar task to locating text search results on a web page. In modern web browsers such as Google Chrome or Mozilla Firefox, there is a functionality to alter the appearance of pieces of text that is the search result. Figure 8 shows screenshots of this functionality in Google Chrome (above) and Mozilla Firefox (below).

Mimicking this functionality, the software implementation is designed to assign visual clues to annotated HTML elements by altering their appearances. Since the annotated HTML elements may not be only text, the visual clue cannot simply be the change in background colour. It is by the author’s intuition that the visual clue should be a small icon next to the element. This is in a similar fashion to a superscript used to indicate an existence of a footnote in writing.

For the second step, an intuitive way would be visualising the links with lines connecting the semantic markup data to the elements. However, this method is non-trivial. This is because the external model is displayed in a tree structure while HTML elements can be scattered around the web page. Some annotated HTML elements may be located outside the portion of the web page the user is viewing. Moreover, judging from the author’s experience in programming, overlaying a web page with graphics is a non-trivial task.

In the author’s opinion, mapping decisions are usually made by the user one at a time. In the state-of-the-art tool, CogZ, the user has to select a term from both ontologies and click on an icon to create one mapping correspondence. In this way the semantic markup mapping tool is designed to visualise only one link between semantic markup and an annotated HTML element at a time.



e Knowledge and Data Engineering Group (KDE

“Researching the challenges of knowledge discovery, representation, engineering and application in next generation application environments”

KDEG investigates the fundamental chall
practical applications of knowledge-driven s
research combines knowledge discovery, repr

The Knowledge and Data Engineering Group (KDEG) | Knowledge and Data Engineer…
File
Edit
View
History
Bookmarks
Tools
Help
The Knowledge and Data Engineering Group …
kdeg.cs.tcd.ie
TCD Home
Faculties & Schools
Courses
Research
Services

he Knowledge and Data Engineering Group (KDEC

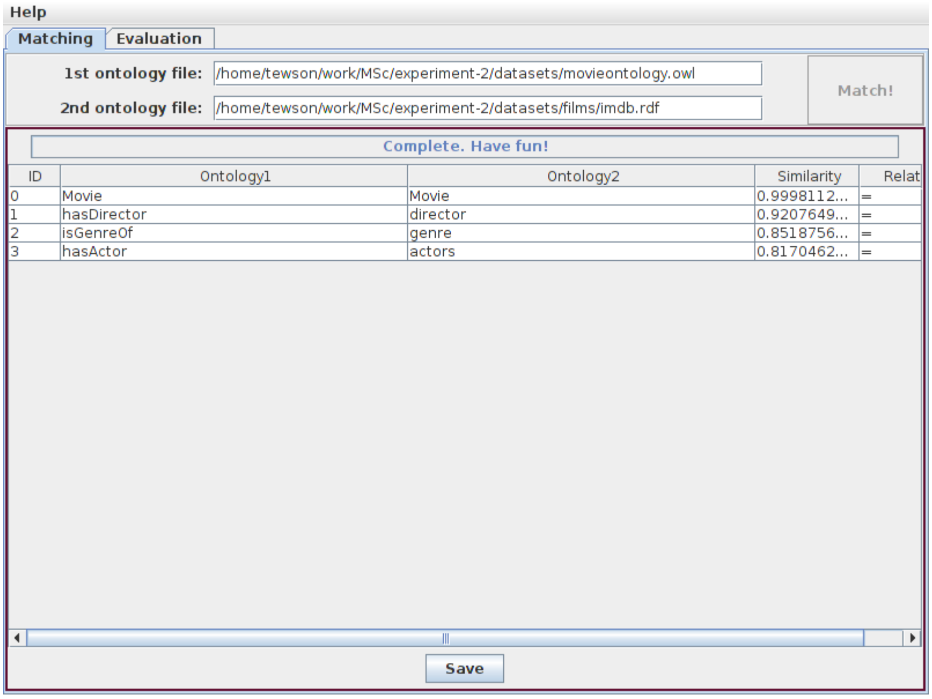
“Researching the challenges of knowledge discovery, representation, engineering and application in next generation application environments”

KDEG investigates the fundamental challer
practical applications of knowledge-driven sys
research combines knowledge discovery, repres
and reasoning with web-based data manager
intelligent systems engineering. This res

Figure 8: Text Search Result Highlighting in Web Browsers

Mapping Correspondence Display

In the Alignment format, a mapping correspondence consists of two terms from two models, a confidence measure value and a relationship type. Falcon (Hu et al. 2008), a state-of-the-art automated mapping tool that has a graphical user interface, displays mapping: correspondences in a tabular format. Falcon's mapping correspondence table is shown in Figure 9. The author of the thesis finds that the fixed number of components in each mapping correspondence makes the tabular display suitable. Thus, the semantic markup mapping tool is designed to display mapping correspondences in a table.



1st ontology file: /home/tewson/work/MSc/experiment-2/datasets/movieontology.owl

2nd ontology file: /home/tewson/work/MSc/experiment-2/datasets/films/imdb.rdf

Complete. Have fun!

ID	Ontology1	Ontology2	Similarity	Relat
0	Movie	Movie	0.9998112...	=
1	hasDirector	director	0.9207649...	=
2	isGenreOf	genre	0.8518756...	=
3	hasActor	actors	0.8170462...	=

Figure 9: Falcon's Mapping Correspondence Table

5.2.4 *SUMMARY*

This section has discussed the design of the semantic markup mapping tool. The design is based on a set of use cases derived from the cognitive support framework introduced in the project CogZ. The designed user interface of the software implementation consists of three components: model display, visual clues and mapping correspondence display. In the next section, an implementation of the design as a software prototype is presented and discussed.

5.3 IMPLEMENTATION

This section presents a software prototype implementation of the semantic markup mapping tool. Sub-section 5.3.1 describes the architecture of the software prototype. Sub-section 5.3.2 introduces the user interface of the software prototype. Sub-section 5.3.3 describes in detail how the requirements were implemented. Finally, sub-section 5.3.4 summarises the implementation. Source code of the software prototype can be found in the accompanying DVD.

5.3.1 ARCHITECTURE

As discussed in section 5.2, the software prototype is required to have access to both semantic markup data on a web page and HTML elements. The author of the thesis chose to exploit the ability of web browsers to access HTML elements on web pages. Among the most used web browsers[60], Google Chrome and Mozilla Firefox stand out as a suitable development platform. This is due to their large communities of extension developer. Since Google Chrome is more popular than Mozilla Firefox[61] and the author is more familiar with Google Chrome, Google Chrome was chosen as the platform for development. The entire prototype was written in HTML5, CSS and JavaScript.

Figure 10 illustrates the software architecture of the software prototype. The semantic markup mapping tool contains an extractor for semantic markup data. At the time of the implementation, the most prominent semantic markup was RDFa. Therefore, RDFa was chosen as the markup for demonstration. The Alignment API (Euzenat 2004) was chosen as the automated matcher. It was configured to use the string edit distance measurement as a typical initial matching function.

[60] Web Browser Market Share Trends, http://www.w3counter.com/trends, Last visited: March 2013.

[61] Ibid.

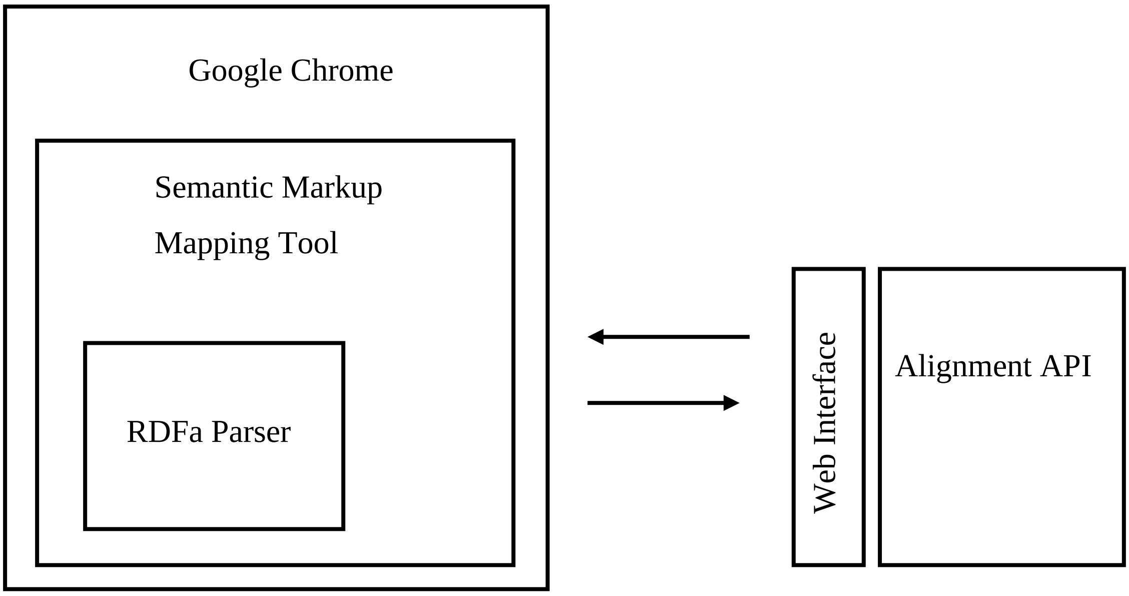
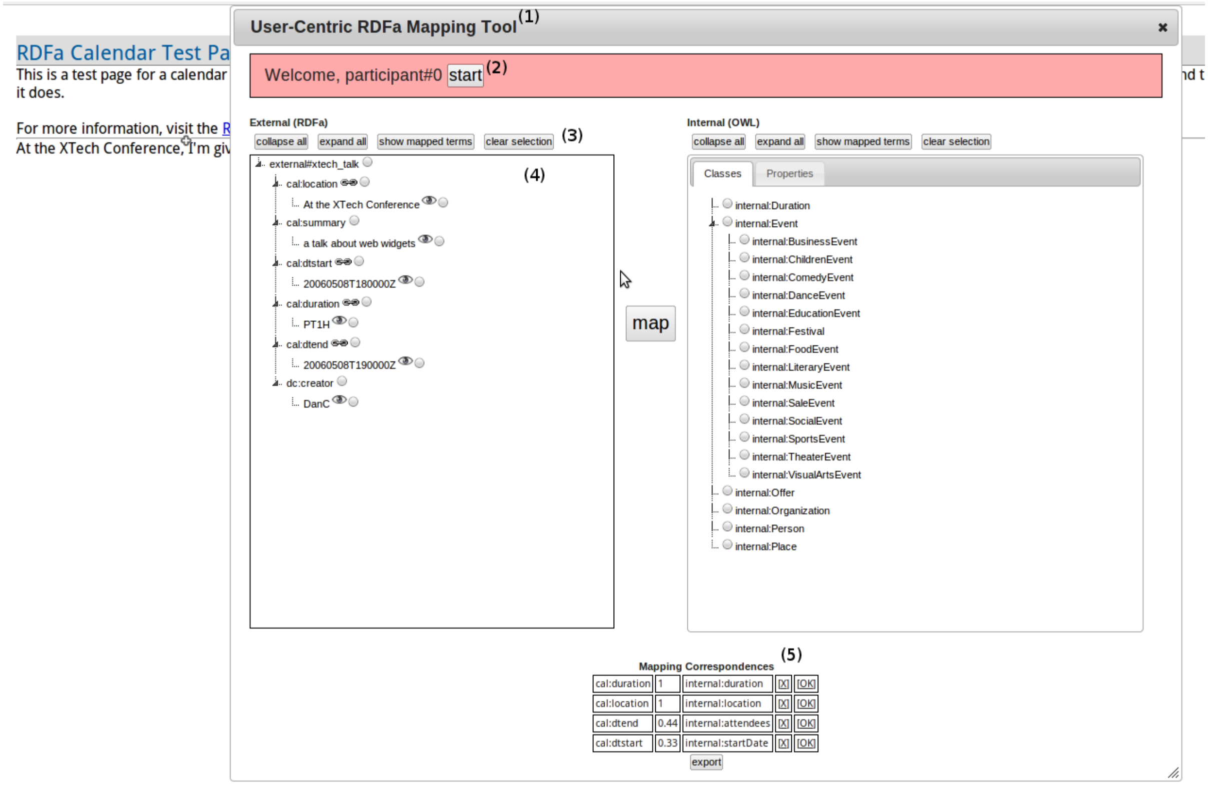


Figure 10: Software Architecture of the Semantic Markup Mapping Tool

To allow the software prototype to request mapping correspondences, a web interface was written in Java Server Pages or JSP. It acts as an intermediate between web requests from the prototype and the Alignment API.

5.3.2 USER INTERFACE

The user interface of the software prototype was created to address the use cases described in the design section. A minor additional component was added to facilitate later evaluation. The numbered parts in Figure 11 are described as follows:



Mapping Correspondences

cal:duration	1	internal:duration	[X]	[OK]
cal:location	1	internal:location	[X]	[OK]
cal:dtend	0.44	internal:attendees	[X]	[OK]
cal:dtstart	0.33	internal:startDate	[X]	[OK]

Figure 11: User Interface of the Software Prototype

1. **Main dialogue box** surrounds all the other components. It appears after the user navigates to a web page with the web browser. After that, the semantic markup data is automatically extracted using the RDFa parser. The user is then presented with a list of internal models. After an internal model is chosen, other components are shown. The main dialogue box can be hidden and displayed if the user wants to take a full view of the web page.

2. **Evaluation dialogue** informs experiment participants acting as users about specific tasks they are asked to do. This evaluation will be discussed in the next chapter.

3. **Model display controls** are shortcuts for mass manipulation of the tree displays. They consist of "collapse all", "expand all", "show mapped terms" and "clear selection". The button "collapse all" hides items in the deeper hierarchy. The button "expand all" shows every term in the tree. The button "show mapped terms" highlights terms that have been assigned a correspondence. The button "clear selection" undoes any selections of terms in both models.

4. **External and internal models** are the inputs to the mapping process. The external model is constructed from semantic markup data on the web page. It is displayed as a hierarchy of subjects, properties and objects. The internal model is constructed from an RDF graph chosen by the user. Internal models are assumed to have more terms

with more expressive definitions. Therefore, the internal model is displayed as hierarchies of classes and properties.

Two kinds of icons can be found after individual terms. An eye icon indicates that a term has a linked HTML element on the web page, or the corresponding visual content. Clicking on the icon will highlight that HTML element. A chain icon indicates that a term has already been mapped to one or more terms in another model. Clicking on this icon will highlight the mapped terms.

5. **Mapping correspondences** are listed as a table. The table is initialised by the Alignment API. The user then can alter the table until it satisfies the mapping task.

The result correspondence table can be exported for use in another application. In this prototype it can be exported as a JavaScript Object Notion (JSON)[62] object, which can be converted into the Alignment format without losing any information.

5.3.3 Requirement Implementation

Model Navigation

The software implementation uses an open source JavaScript library called rdfQuery[63] for transforming RDFa data to RDF triples. rdfQuery also reads RDF files into a triple store.

Candidate Mapping Correspondence Manipulation

The Alignment API, the matcher chosen in this implementation, not only provides a console interface, but also provides Java classes that can be extended. This allowed the author to develop a small wrapper written in JSP. The wrapper takes requests from the prototype and uses the Alignment API classes to generate results and send them back.

The Alignment API also allows implementation of matching algorithms. More sophisticated matching such as structure analysis can be added in the future with minimum change in the mapping tool.

Mapping Correspondence Generation

Mapping decisions made by the user is stored in a triple store in the memory until exported.

[62] JSON is a simple key-value data structure based on the JavaScript language. It was chosen by the author because the software prototype was written in JavaScript.

[63] rdfQuery, http://code.google.com/p/rdfquery, Last visited: January 2013.

Mapping Result Export

Many semantic matchers share a common mapping result format because of the Ontology Alignment Evaluation Initiative (OAEI)[64], an initiative focusing on evaluating different semantic matching techniques. The format used in OAEI is called the Alignment format, which is also used in the Alignment API.

The Alignment format is an RDF-based format. It contains data on the matching algorithm, mapping input URIs and mapping correspondences. Each correspondence consists of the URIs of the matched terms, the type of relation and a confidence value of the match.

In this work, the mapping results can be exported in the JSON format. The JSON format, being a simple data structure, can be converted into the Alignment format without losing any data.

5.3.4 SUMMARY

In this section, the implementation of the semantic markup mapping tool was discussed. The tool was written with the same technologies as ordinary web pages. The prototype tool supports only one semantic markup and one automated matcher. However, the semantic markup extractor and the matcher were designed to be modular. Therefore, new semantic markup extractors and matchers can be added to the software prototype with minimal modification.

5.4 CHAPTER SUMMARY

In this chapter, a semantic markup mapping tool was presented. The semantic markup mapping tool was designed to exploit the links between semantic markup data and its annotated HTML elements to reduce human effort in a mapping task. A design of the tool was presented with 12 use cases derived from the cognitive support framework. An implementation of the tool was done as a Google Chrome web browser extension.

In the next chapter, an evaluation of the software prototype is discussed.

[64] Ontology Alignment Evaluation Initiative, http://oaei.ontologymatching.org/, Last visited: September 2012.

6 VISUAL CONTEXT IN SEMANTIC MARKUP MAPPING: EVALUATION

In this chapter, the evaluation of the semantic markup mapping tool is presented. A user experiment was conducted with participants with variety in experience with semantic web technologies. The rest of this chapter consists of the objectives, hypothesis, design, setup and results of the experiment. The results are then analysed. This chapter concludes with a summary that leads to another phase of the work in this thesis.

6.1 OBJECTIVES

This evaluation was conducted to:

- Observe and measure the performance in mapping tasks performed by a variety of sample users,
- Measure the ease of use of the semantic markup mapping tool perceived by a variety of sample users, and
- Identify possibilities for improvement of the semantic markup mapping tool.

6.2 HYPOTHESIS

This experiment was planned to test the following hypotheses:

A. Different groups of users perform mapping tasks with similar performance, consisting of accuracy and speed.

B. Different groups of users find the software prototype intuitive to use.

C. Users find exposing links between semantic markup data and corresponding visual elements helpful in semantic mapping.

6.3 EXPERIMENT DESIGN

This section discusses the design of the experiment. First, sub-section 6.3.1 describes an overview of the experiment. Then, sub-section 6.3.2 describes the participants. After that, sub-section 6.3.3 lists the tasks used in the experiment. Subsequently, sub-section 6.3.4 describes how data was collected. Finally, subsection 0 presents the questionnaire.

6.3.1 Overview

The experiment consisted of asking each participant to perform various tasks with the tool following the instructions on printed instruction sheets. The instruction sheets can be found at the appendix of this thesis. While the participants were following the instructions, time they spent on tasks and mapping correspondences they generated were collected in the background. They were subsequently asked to give feedback in a short post-experiment questionnaire, which is also included at the appendix.

6.3.2 Participant Categorisation

As this experiment aims to measure the tool's ease of use, participants were classified into categories described in Table 1. This was done in order to determine whether using the tool depends on the user's level of understanding in semantic web technologies, or at least on the user's familiarity in using software in general. The participants were classified based upon answers from a questionnaire related to their experience, and brief interviews with the experimenter.

6.3.3 Task Design

The participants were first given an up to 20-minute introduction to RDF, RDFa and semantic mapping. They were also given a manual of the tool's user interface. The participants then were allowed to familiarise themselves with the tool for up to 10 minutes. After that, the participants began to perform tasks from two categories: *navigation* tasks and *mapping* tasks. The categorisation of the tasks was designed to reflect the increasing level of understanding that a user must have in order to perform a full mapping task. In other words, the user was expected to first be able to navigate through the tool's user interface. The user then should be able to use the tool's user interface to perform mapping tasks without making decisions, i.e., guided mapping tasks. Finally, the user should be able to perform mapping tasks themself.

Table 1: Participant Categorisation

Category	Description
Domain experts	People who are studying, or have worked on, semantic web technologies. Participants in this group have used semantic web software and are familiar with technical terms.
IT personnel	People who have little or no experience with semantic web technologies but are studying or have worked in areas of information technology. Participants in this group represent general workers in the semantic web tool chain. They may not be familiar with semantic web technical terms. However, they are assumed to be adept in using non-generic software.
End users	People who are end users and familiar with web browsing. This group were intended to explore a possibility of implementation of the method in a more casual environment.

This sequencing was designed to help determine later during analysis where the user be confused or where the user might misunderstand the process. There are subtasks in each category as follows:

1. **Navigation**
   - Navigation#1: Locating a piece of information in extracted RDFa data
   - Navigation#2: Locating a piece of information in extracted RDFa data from its corresponding visual content
   - Navigation#3: Answering a question on structure of an internal model
   - Navigation#4: Answering a question about machine-generated mapping
2. **Mapping**
   - Mapping#1: Producing a mapping correspondence table as instructed
   - Mapping#2: Producing a mapping correspondence table as instructed for a web page embedded with more complicated data
   - Mapping#3: Producing a mapping correspondence table independently
   - Mapping#4: Producing a mapping correspondence table independently for a more complicated web page

6.3.4 Data Collection

To ensure that participants would perform the tasks in the most natural manner as possible, details on data collection were not disclosed to them. There was a timer recording time that was spent on each task as a measure of difficulty. For navigation tasks, answers were recorded for later analysis in order to identify confusion in using the tool. For independent mapping tasks, mapping correspondences were recorded for analysis of how participants understand semantic mapping and to what extent they felt satisfied with the results. There was a maximum time of ten minutes for the last two tasks to limit the total time for one session not to exceed one hour. This was to maximise the number of participants while retaining enough time for all the tasks. Participants were allowed to stop at any time they wanted.

6.3.5 Post Experiment Questionnaire

After finishing all tasks, participants were asked to complete a questionnaire through a web form that let them evaluate usability of the software. They were also asked to point out features that were particularly useful and those that were confusing. The questions are listed as follows:

1. Familiarity with web browsing
    - This was to verify whether using the tool depends on experience in web browsing.
    - Question: "How would you rate your familiarity in using the internet through web browsers?"
    - Input: 1 (unfamiliar at all) – 5 (most familiar)
2. Knowledge in semantic web technologies
    - This was to verify whether prior knowledge in semantic web technologies help a user use the tool.
    - Question: "How would you rate your knowledge in semantic web technology in general?"
    - Input: 1 (none) – 5 (very competent)
3. Usability of the software prototype
    - This was to measure the perceived usability of the tool by the users.
    - Question: "How would you rate the overall usability of the prototype?"
    - Input: 1 (obstructively complicated) – 5 (very useful)
4. Tree display interface
    - This was to evaluate whether the choice of using a tree display instead of a graph display helps simplify mapping tasks.
    - Question: "How would you rate the appropriateness of the tree display?"
    - Input: 1 (most inappropriate) – 5 (most appropriate)

5. Tree navigation
    - This was to evaluate whether the functionality of hiding parts of RDF tree representation is helpful or confusing.
    - Question: "How would you rate the appropriateness of the collapsible tree navigation?"
    - Input: 1 (most inappropriate) – 5 (most appropriate)
6. Useful features
    - This was to let the participants pick out what they thought was the most useful feature.
    - Question: "What is the most useful feature of the prototype?"
    - Input: Arbitrary names of components in the software prototype.
7. Confusing features
    - This was to determine what parts of the software caused confusion.
    - Question: "What is the most confusing feature of the prototype?"
    - Input: Arbitrary names of components in the software prototype.
8. Suggestion for improvement
    - This was to let the participants comment on what is missing from the software prototype that would make it easier to use.
    - Question: "Please suggest features that would improve the prototype."
    - Input: Arbitrary comments.

6.4 EXPERIMENT SETUP

This section describes how the experiment was executed.

6.4.1 PARTICIPANTS

Participants were recruited through internal mailing lists of the School of Computer Science and Statistics, Trinity College Dublin. The author also recruited some participants from a private company with which he had links.

15 people responded. 7 were categorised as "domain experts", 4 were categorised as "IT personnel" and the other 4 "end users".

6.4.2 ENVIRONMENT

Only one participant was allowed per session. In this way the experimenter could fully observe their interaction with the tool. The participant was placed in front of a desktop monitor, a mouse and a keyboard connected to a dell laptop computer with 2.53GHz CPU and 4GB RAM. Google Chrome web browser, in which the tool is installed, is the only software application exposed to participants.

The experimenter sat beside the participant and answered questions regarding the user interface, not the actual tasks, that might arise.

6.4.3 DATA RESOURCES

In each task, the participants worked with different web pages and internal models to capture their candid performance[65]. The dataset is described as follows:

- In navigation tasks, participants worked on a simple event calendar web page that contains seven triples.
- In Mapping#1, they were given a web page describing personal contact data that contained 31 triples. The internal model contained 37 classes and 55 properties.
- In Mapping#2, they were given a web page of a department store that contained 77 triples. The internal model contained 12 classes and 26 properties.
- In Mapping#3, they were given a web page of a music album review that contained 36 triples. The internal model contained one class and 16 properties.
- In Mapping#4, they were given a web page of an electronic accessory product that contained 121 triples. The internal model contained one class and 11 properties.

6.5 RESULTS AND ANALYSIS

This section presents results from the experiment and analysis on the results. The results are grouped as navigation tasks, guided mapping tasks, independent mapping tasks and usability questionnaire. Results from the navigation tasks should reflect mainly on usability of the software prototype. Results from the guided mapping tasks should reflect on how experience in semantic web technologies affects usage of the software. Answers the participants gave in the questionnaire should address overall perception of the tool and possibly unthought-of issues.

6.5.1 NAVIGATION TASKS

The following results were collected from the navigation tasks designed to evaluate the speed of the participants' interaction to the software prototype. An analysis on the results is presented subsequently.

Results

For navigation tasks, the mean of time spent by each participant group on each task is shown in Figure 12. The median values are shown in Figure 13.

[65] The dataset can be found in the accompanying DVD.

There were several incorrect answers from each group. Table 2 describes where they occurred and how many of them did. Their causes are listed as follows:

- **Domain experts**
    - Choosing a property instead of its object (1 occurrence in task Navigation#1)
    - Choosing an object instead of its property (2 occurrences in task Navigation#2)
- **IT personnel**
    - Choosing an object instead of its property (2 occurrences in task Navigation#2, 2 occurrences in task Navigation#3 and 1 occurrence in task Navigation#4)
    - Not using machine-generated mapping table when instructed so (1 occurrence in task Navigation#4)
- **End Users**
    - Choosing an object instead of its property (1 occurrence in task Navigation#1)

Table 2: Percentage of Incorrect Answers for Navigation Tasks

Participant category	Task number			
	#1	#2	#3	#4
Domain experts	14.28	28.57	0.00	0.00
IT Personnel	0.00	50.00	50.00	50.00
End users	25.00	0.00	0.00	0.00
Overall	13.33	26.67	13.33	13.33

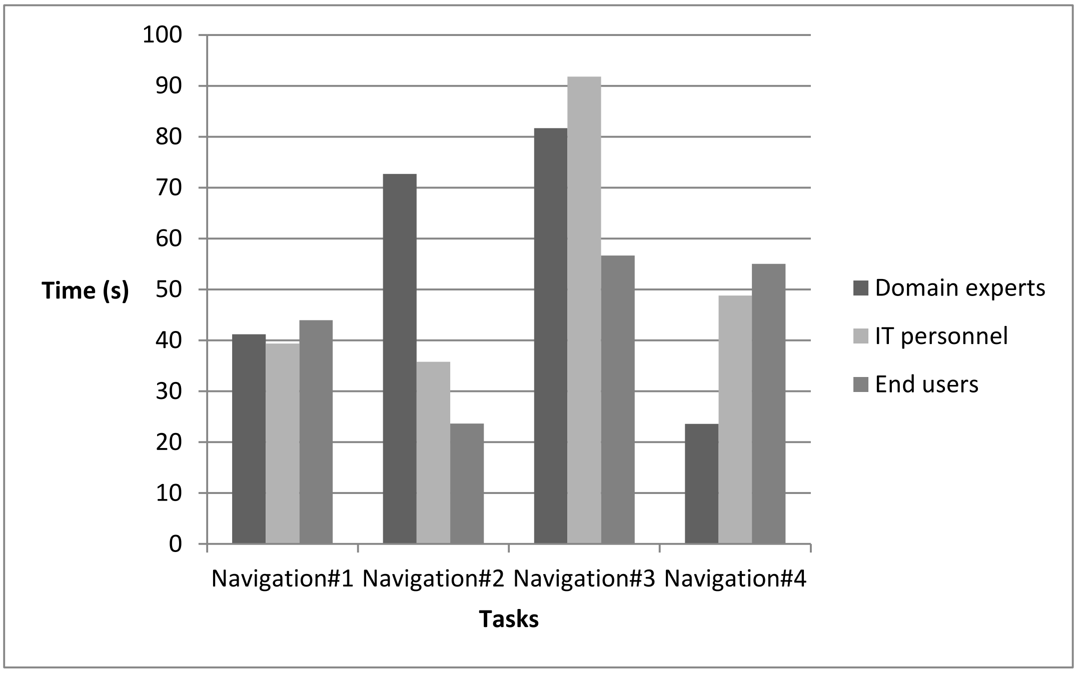


Figure 12: Mean Time Spent in Navigation Tasks

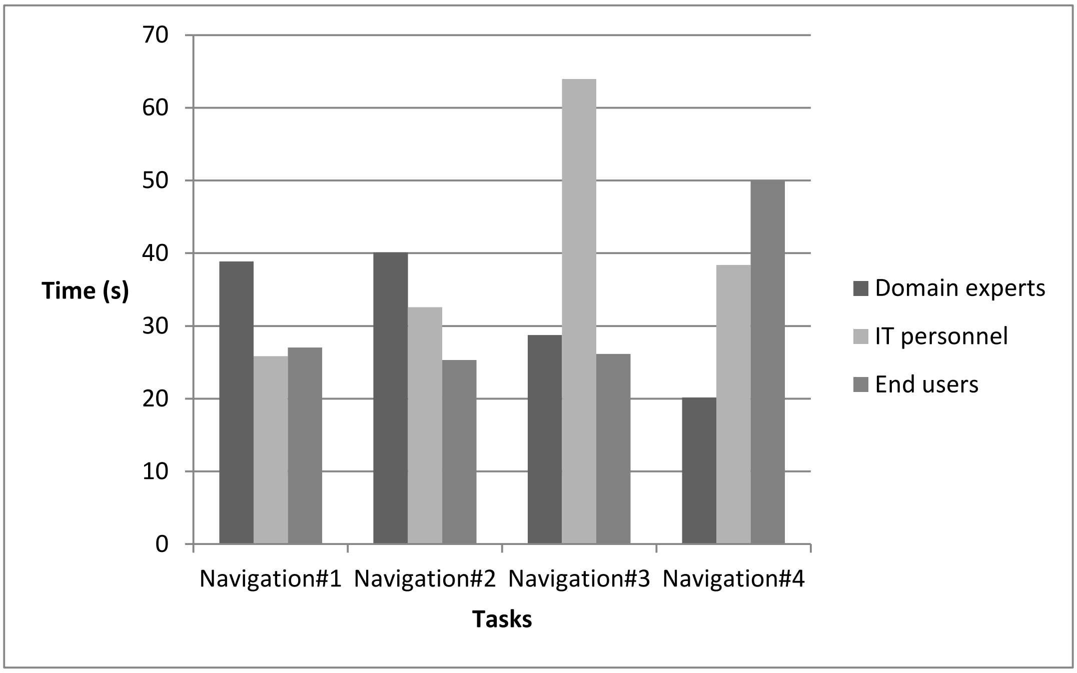


Figure 13: Median Time Spent in Navigation Tasks

Analysis

Mean time spent in navigation tasks by each group is shown in Figure 12. To address issues from outliers, median values are shown in Figure 13. Incorrect answers received in each tasks are summarised in Table 2.

The first task, "**Navigation#1**", was intended to be the easiest one. It required the participants to look at one of the RDF trees. Having been briefed earlier, they should understand how the tree hierarchy shows relationships between terms as a subject, a property or an object. The participants should find an answer to the question as an object of one triple and submit its text. This task resulted in no significant difference in mean time spent by all groups of participants. Incorrect answers received in this task were minor, and these suggested that there was some confusion between properties and their objects.

The second task, "**Navigation#2**" required the participants to read the web page content. Then, they were asked to locate a property in a triple associated to a particular piece of text in the content. Contrary to the experimenter's assumption, this task took several domain experts longer than the rest of the participants. From the experimenter's observation, this was because some domain experts were not sure whether the "property" in question was meant as a triple's property and took their time to decide. However, as shown in Figure 13, the median of the results from this task shows less significant difference among the groups. This means the few outliers in the domain expert group could be dismissed as errors. Confusion between properties and objects still existed in this task. This issue will be addressed later on in the questionnaire results.

For the third task, "**Nagivation#3**" the participants were asked to locate a property of a triple that links the given subject and the given object. The difference between mean time and median time spent among each group again suggests an effect of outliers. The median result from domain experts in this task is relatively lower than the previous ones. This shows that they possibly had got used to the experiment. Meanwhile, end users showed no significant improvement. On the other hand, the cause of the spike in the result from IT personnel is difficult to determine because of the small sample size.

In the last task in this category, "**Navigation#4**", machine-generated mapping correspondence table was introduced. The participants had to answer a question on information from the table. We can see that domain experts spent relatively much less effort in locating that information. This could be explained by their prior experience. The results showed that the mapping correspondence table was unusual to the rest of the participants. It took them longer and they made more mistakes.

6.5.2 *GUIDED MAPPING TASKS*

Results

Mean time spent on the first two mapping tasks with mapping guidelines is shown in Figure 14.

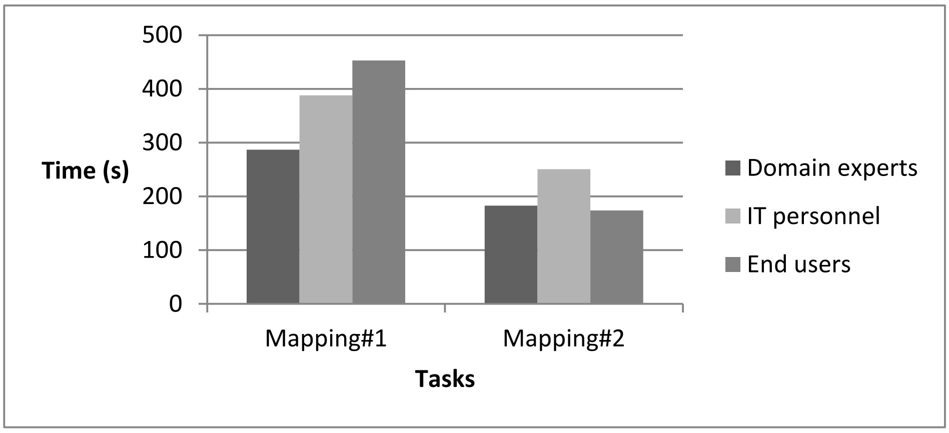


Figure 14: Mean Time Spent in Guided Mapping Tasks

Analysis

When it comes to getting familiar with a mapping task, it can be seen from Figure 14that all groups quickly become familiar with the tool. Domain experts had 36.30% decreases in time spent undertaking a task. IT personnel had 35.45% and end users 61.63%. These improvements occurred despite the task Mapping#2 being more complicated than Mapping#1 in terms of number of triples.

In the task Mapping#1, the performances were ranked from domain experts to end users. This was possibly because domain experts were supposed to be more familiar with navigating semantic data. IT personnel were supposed to be familiar with structured data in general. Some errors, however, occurred when some participants misread the mapping guideline and needed more time to correct the results.

6.5.3 *INDEPENDENT MAPPING TASKS*

Results

For the last two tasks of independent mapping, an observation on quality of mapping correspondences is described in Figure 15

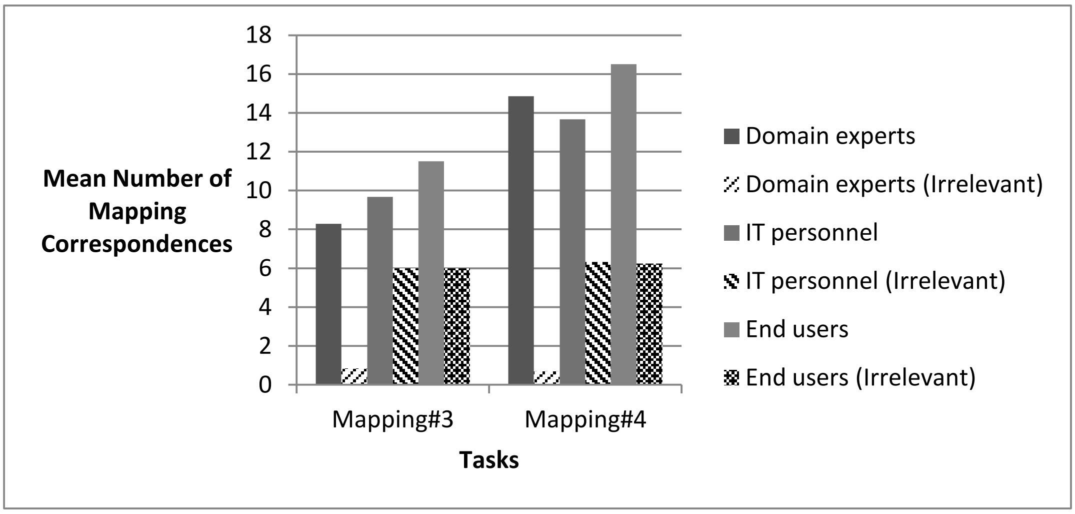


Figure 15: Mean Numbers of Total and Irrelevant Correspondences Created by Participants

Data in Figure 15 consists of mean numbers of mapping correspondences each participant used the tool to generate. These numbers are accompanied with mean numbers of 'irrelevant' correspondences that are defined in Table 3.

Table 3: Percentages of Types of Irrelevant Correspondences in Independent Mapping Tasks

Description	Domain experts		IT personnel		End users	
	#3	#4	#3	#4	#3	#4
Mapping a literal value to a property	83.33	60.00	94.44	94.74	25.00	40.00
Mapping between terms that should have a superclass-subclass relationship	N/A	40.00	5.56	5.26	4.17	8.00
Mapping terms that do not have lexical similarity	16.67	N/A	N/A	N/A	70.83	52.00

Analysis

After the experience of guided mapping, participants proceeded to tasks more resembling actual semantic mapping. Support for the hypothesis can be seen in Figure 15 and Table 3. Participants who have background in semantic web technologies created far fewer *irrelevant* mapping correspondences than the rest did.

As described in Table 3, irrelevant mapping correspondences are categorised as:

- **Mapping a literal value to a property**:
  This is a common mistake that occurs in every group. It was possibly done in a similar fashion to filling out a form.
- **Mapping between terms that should have a superclass-subclass relationship**:
  In a similar trend to mapping objects to properties, subclasses are sometimes mistakenly mapped to superclasses. It might be explained by how people read and understand that two terms are related. These mappings are not entirely incorrect. They would have been relevant if the mappings were labelled 'isPropertyOf' or 'isSubclassOf'. Unfortunately in this case the participants sometimes neglected the instruction on mapping terms representing similar concepts.
- **Mapping terms that do not have lexical similarity**:
  It was observed that this case resulted from too broad an interpretation of terms, e.g., mapping 'Summary' to 'About' or 'copyright' to 'author'.

From Figure 15, it can be clearly seen that domain experts rarely created irrelevant correspondences. The rest of the participants made similar amounts of irrelevant correspondences that comprise approximately half of the total. However, according to Table 3, irrelevant correspondences made by the IT personnel group were more justifiable. They were mostly related terms. Those from end users, on the other hand, contained more arbitrary errors.

On average, end users generated the most number of total correspondences. However, the numbers vary significantly among individuals. The trend in this group was that participants who created a large number of mapping correspondences were usually the ones who generated more irrelevancies. This possibly reflects on inconsistent understanding of the topic among them. It also occurred among participants from the IT personnel group.

6.5.4 USABILITY QUESTIONNAIRE

Results

From the questionnaire, mean usability scores (out of five) of the whole tool and two main navigation interface components given by each group are described in Table 4.

Table 4: Mean Usability Score

Component	Domain experts	IT personnel	End users	Average
Overall	4	3.5	3	3.6
Tree Displays	3.86	3.75	4	3.87
Collapsible Navigation	4.29	4	4.75	4.33

Several features of the tool were selected to be the most useful ones and the most confusing ones. These are described in Table 5 and Table 6 accordingly.

Table 5: Most Useful Features

Feature	Domain experts	IT personnel	End users	Total
Interactive linking between visible web page contents and their corresponding triples	3	3	2	8
Search functionality	2	0	0	2
Indication of a term having been mapped to another term	1	0	1	2
Machine-generated mapping	1	0	0	1
Displaying two models side-by-side	0	0	1	1
Separation of classes and properties in the internal model	0	1	0	1

Table 6: Most Confusing Features

Feature	Domain experts	IT personnel	End users	Total
Level of details in models	3	1	0	4
Difference in model displays	1	0	1	2
No separation of different roles in external model	1	0	0	1
Collapsible trees	0	1	0	1
No human-readable labels for terms	1	0	0	1
Mapping in general	1	0	0	1

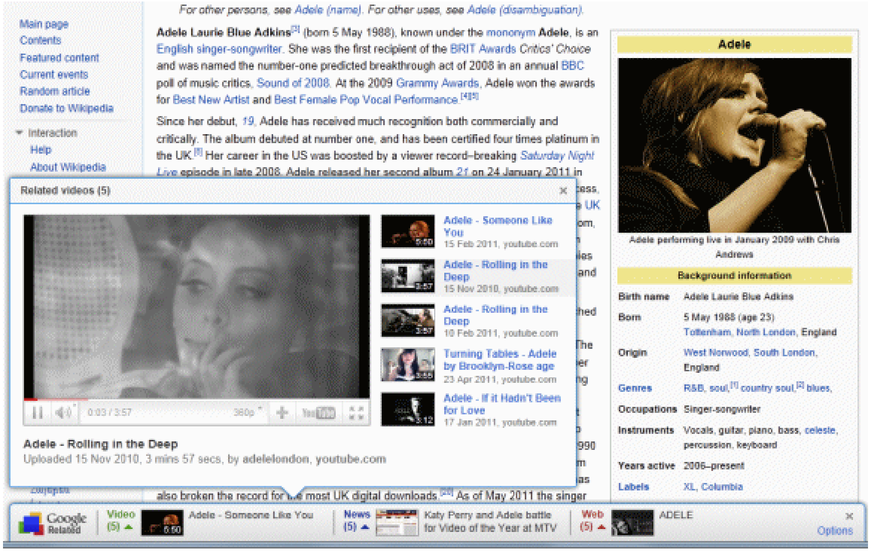
When asked to suggest features or improvement that would make the tool easier to use, participants responded in the following way:

- Domain experts
    - Less obstructive user interface, i.e., not having the tool to take up most screen space
    - Clearly separating subjects, predicates and objects in triples; possibly using a table
    - Built-in examples
    - Better search functionality, i.e., being able to search for collapsed terms
    - Removal of duplicate terms
- IT personnel
    - Prevention of mapping terms of different roles
    - Colour coding for different roles
    - Better search functionality
    - Built-in examples
- End users
    - Pop-up help dialogues for each feature
    - Simpler terminology

Analysis

Overall evaluation as indicated in Table 4 seems to be acceptable. It is not surprising that the tool as a whole was more appealing to domain experts than the rest as they tend to be more familiar with semantic web tools or have more understanding of how they work.

The ability to relate visual content and semantic data was most appreciated. It could be taken a step further by making the tool take up less screen space. An example of such a user interface approach is an extension for Google Chrome web browser called ‘Google Related’ that stays on the bottom of the page (see Figure 16).



For other persons, see Adele (name). For other uses, see Adele (disambiguation).

Adele Laurie Blue Adkins[2] (born 5 May 1988), known under the mononym **Adele**, is an English singer-songwriter. She was the first recipient of the BRIT Awards *Critics' Choice* and was named the number-one predicted breakthrough act of 2008 in an annual BBC poll of music critics, Sound of 2008. At the 2009 Grammy Awards, Adele won the awards for Best New Artist and Best Female Pop Vocal Performance.[4][5]

Since her debut, *19*, Adele has received much recognition both commercially and critically. The album debuted at number one, and has been certified four times platinum in the UK.[6] Her career in the US was boosted by a viewer record–breaking *Saturday Night Live* episode in late 2008. Adele released her second album *21* on 24 January 2011 in

also broken the record for the most UK digital downloads.[20] As of May 2011 the singer

Adele - Someone Like You
15 Feb 2011, youtube.com

Adele - Rolling in the Deep
15 Nov 2010, youtube.com

Adele - Rolling in the Deep
10 Feb 2011, youtube.com

Turning Tables - Adele by Brooklyn-Rose age
23 Apr 2011, youtube.com

Adele - If it Hadn't Been for Love
17 Jan 2011, youtube.com

Adele - Rolling in the Deep
Uploaded 15 Nov 2010, 3 mins 57 secs, by adelelondon, youtube.com

Adele - Someone Like You

Katy Perry and Adele battle for Video of the Year at MTV

ADELE

Adele

Adele performing live in January 2009 with Chris Andrews

Background information	
Birth name	Adele Laurie Blue Adkins
Born	5 May 1988 (age 23) Tottenham, North London, England
Origin	West Norwood, South London, England
Genres	R&B, soul,[1] country soul,[2] blues,
Occupations	Singer-songwriter
Instruments	Vocals, guitar, piano, bass, celeste, percussion, keyboard
Years active	2006–present
Labels	XL, Columbia

Figure 16: User Interface of Google Related

The search functionality greatly helped speed up the guided mapping tasks. However, it still needs to be improved to be able to search for terms that are hidden in the tree.

Tree display was perceived to be not very useful due to its lack of distinction among terms and differences among trees themselves. Being pointed out as the most confusing feature in Table 6, the tree display needs several aspects of improvements. Taking into account the list of recommended features, the ideal model display would have clear labels, probably colour-coded, of roles of terms, e.g., subjects and properties. This may however discourage mapping of terms that have different roles.

Help functionality and more examples were also requested. Perhaps briefing time longer that around 20 minutes used in the experiment would be more helpful in the real tasks.

There is a possibility of a new approach in mapping that is more like content annotating. One may imagine all visual content being highlighted and attached with clickable buttons (similar to cross icons in the current version). When buttons are clicked, users will be able to see semantic data for that element and browse through the internal model to find terms to map. However, this approach will discourage users from performing mapping on invisible content, which may still need the tree display approach.

6.6 Conclusions

From the analysis undertaken, conclusions can be drawn for each hypothesis as follows:

Hypothesis A: Different groups of users perform mapping tasks with similar performance, consisting of accuracy and speed.

The results show that; overall, domain experts still perform with best accuracy and fastest time in mapping tasks. For the rest of the participants, the results fluctuated with large differences. All groups of participants have the most similar performance when the mapping task is guided. This could mean that if more preparation was allowed and more guidance was provided, the participants would have performed better.

Background knowledge in information technology did not prove to be helpful to participants within the respective group.

Hypothesis B: Different groups of users find the tool intuitive to use.

The tool was rated as somewhat usable at the mean score of 3.6 out of 5. It was the domain expert group that gave the highest rating.

The model display was found to be the most confusing component. Suggestions were made for the software prototype to make subjects, properties and objects more distinctive from each other. For example, in addition to hierarchy display, terms could be colour-coded.

Hypothesis C: Users find exposing links between semantic markup data and annotated HTML elements helpful in semantic mapping.

Visualisation of links between semantic markup data and the annotated HTML elements was rated as the most useful feature of the tool. It did not prevent participants to make the most common mistake: mapping literal values to properties. However, this mistake could be viewed as more forgivable than the others. This is because almost all results with this mistake would have been otherwise correct if the properties were mapped instead. This indicates that at least the participants associated two properties correctly. The mistake might result from the lack of complete understanding of the mapping task.

The future work for the semantic markup mapping tool is to implement suggested features. Also, a larger number of participants in an evaluation would minimise fluctuations in experiment results.

7 DOM-Based External Model Enrichment: Design and Implementation

This chapter presents a process of expanding the external model to increase precision and recall of results of the mapping process. First, section 7.1 discusses the motivation behind the proposed process. After that, section 7.2 describes the design of the process. Furthermore, section 7.3 introduces a prototype software implementation of the process. Finally, section 7.4 summarises the design and implementation of the process.

7.1 Motivation

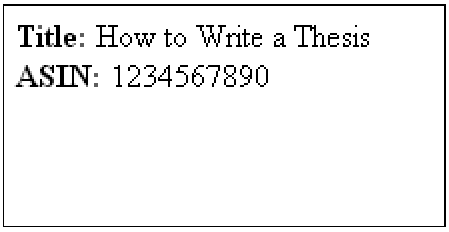
An experiment using semantic markup in semantic mapping was discussed in chapter 6. The experiment explored the manual process of mapping an external knowledge model created from semantic markup data to an internal knowledge model. The mapping process was undertaken by users of different technical backgrounds. In the experiment, the users were given an automatically generated set of mapping correspondences as a guideline. They were also able to relate HTML elements on the web page and semantic markup data they were annotated with. The experiment concluded that links between semantic markup data and the rendered, annotated elements assist the user in performing mapping tasks. This prompts a question of whether additional data from the rendered web page could assist automated mapping as well.

In the scenario of a data aggregation service discussed in chapter 4, the ultimate goal is to extract every piece of data in a web page that the human user indicates as fitting into the internal model. External models are used as recommendations to reduce workload of the human user. These external models usually use terms from third-party vocabularies such as Schema.org. Being created by different authors, it is expected that these third-party vocabularies would not have a similar structure and a similar terminology to the internal model. Therefore, using only semantic markup data in the external model that uses third-party vocabularies may not cover all data on the web page that fits into the internal model.

By observing the document object model (DOM) representation of a web page, it can be seen that there may be other information that the external model based on the semantic markup data does not contain. An HTML code snippet in Figure 17 and the rendered web page in Figure 18 illustrate an example of this situation.

```
<DIV ITEMSCOPE ITEMID = “1” ITEMTYPE = “http://schema.org/Book”>

     <DIV>

          <STRONG>Title: </Strong><SPAN ITEMPROP = “name”>How to
          Write a Thesis</SPAN>

     </DIV>

     <DIV>

          <STRONG>ASIN: </STRONG> <SPAN>1234567890</SPAN>

     </DIV>

</DIV>
```

Figure 17: HTML Code Snippet Containing Microdata and Other Extractable Data



Title: How to Write a Thesis
ASIN: 1234567890

Figure 18: Rendered Web Page from a Snippet in Figure 17

From the snippet, the parent DIV element is annotated with Microdata to be a subject of a triple. The subject is assigned the type of *Book* described in Schema.org. From the rendered web page, a human would perceive that there are two data fields regarding this book: the title of the book and its Amazon Standard Identification Number or ASIN. The snippet shows that the SPAN element that contains the book title is annotated with a property called *name*, also from Schema.org. The other SPAN element containing the ASIN of the book however is not annotated with any property. This is due to the lack of a property related to ASIN in Schema.org.

Let us assume that there is a property called *ASIN* in the internal model. The external model of this example web page that uses semantic markup data alone would not have any property to be mapped correctly to *ASIN*. If the string “ASIN” is extracted from the DOM and added to the external model as a property, it may be mapped to the internal model's *ASIN*. The value "1234567890" can then be extracted by a simple algorithm.

Another example of additional data that can be extracted from the DOM and added to the external model is text strings that share the same meaning with terms from the semantic markup data. For example, from the same snippet, the human user would perceive that the title of the data field containing the book's title is "Title". However, the SPAN element containing the title of the book is marked up with the property *name*. This is because Schema.org uses the property *name* instead of *title* to describe words by which something is known.

Let us assume again that a property called *title* is used in the internal model to describe a book's title. As discussed in section 2.5, many semantic matching algorithms are based on string comparison. Matching the property *name* from the external model to the property *title* from the internal model may not result in the highest possible confidence value. However, if the string "Title" is extracted from the DOM and added to the external model as a property and has the same value as Schema.org's *name,* it may result in a higher confidence value.

With the added data, the external model now contains more terms. In a mapping process that analyses every term from two models, the extra terms are likely to increase the chance of getting more mapping correspondences, some of them probably with higher confidence values. This process of adding data extracted from the DOM to the external model will be referred to from now on as the *external model enrichment process*.

Before discussing further the external model enrichment process, the author of the thesis would like to note that the semantic markup standard that will be used in examples and the implementation is changed from RDFa to Microdata. The following sub-section discusses the reason behind this change.

7.1.1 Microdata as the rising Semantic Markup Standard

There have been new applications in the area of semantic markup data during the time between the experiment presented in chapter 6 and the experiment designed to evaluate the proposed process. It was observed that semantic markup data in the form of RDFa may have lost its prevalence. A recent survey (Mika and Potter 2012) collects lists of 20 websites that publish most semantic markup data in Microformats, RDFa and Microdata. The survey shows that the list of websites that publish RDFa contains less variety than that for Microdata, with ten websites being from the same company: TripAdvisor. On the other hand, the data collected for Microformats was focused only on contact information data.

According to the survey, RDFa exists in the majority of the collected web pages. However, it is shown in the survey that RDFa is used less than Microdata in terms of domain-specific content annotation. Many of the top RDFa namespaces are either Facebook's Open Graph Protocol or non-domain-specific vocabularies such as the RDF namespace or DublinCore[66]. The Open Graph Protocol terms are used only in <META> tags[67].

Microdata, on the other hand, is found by the author of the thesis to be used for content annotation in a variety of domains. Microdata had gained much popularity in the recent years with about 470% increase in the number of URLs it is used in from 2009 to 2012 (Mühleisen and Bizer 2012). In the period of six months from February August 2012, the number of Microdata triples collected by the Web Data Common project rose from 404,413,915[68] to 1,488,063,426[69] triples or by approximately 260%. Yandex[70], the most visited website in

[66] The Dublin Core Metadata Initiative, http://dublincore.org/, Last visited: September 2012.

[67] Facebook, Inc., "The Open Graph Protocol", http://opengraphprotocol.org/, Last visited: September 2012.

[68] Web Data Commons Extraction Report – February 2012 Corpus, http://webdatacommons.org/2012-02/stats/stats.html, Last visited: March 2013.

[69] Web Data Commons Extraction Report – August 2012 Corpus, http://webdatacommons.org/2012-08/stats/stats.html, Last visited: March 2013.

[70] Yandex, http://www.yandex.com/, Last visited: March 2013.

Russia[71] and the third most used search engine in Europe in 2008[72], have also speculated the growth of Microdata. In a presentation at the 11th International Semantic Web Conference (ISWC2012), it was shown that the number of URLs found by Yandex to contain Schema.org terms rose by 2.6% from June 2011 to November 2012[73].

It has been speculated that the growth of Microdata has been caused by an attempt of search companies to promote a new content annotation vocabulary that is Schema.org[74] (Mika and Potter 2012). The author of the thesis has also found a number of websites in the same domains as the chosen internal models that use Microdata.

Differences between Microdata and RDFa lie mainly in the way to embed them in a web page. They both can be converted to RDF graphs. With the greater variety of data available, Microdata was chosen as a representation of semantic markup in the implementation and the associated experiment.

The following section discusses how the external model enrichment process is designed.

[71] comScore Releases Russian Web Site Rankings for February, http://www.comscore.com/Insights/Press_Releases/2008/04/Top_Russian_Web_Sites, Last visited: March 2013.

[72] comScore Releases March 2008 European Search Rankings, http://www.comscore.com/Insights/Press_Releases/2008/05/Top_European_Search_Engines, Last visited: March 2013.

[73] Schema.org Update, http://www.slideshare.net/AlexShubin1/schemaorg-iswc2012-15283142, Last visited: March 2013.

[74] Google, Inc., Yahoo, Inc., and Microsoft Corporation, "What is Schema.org?", http://www.schema.org, Last visited: March 2013.

7.2 DESIGN

This section begins with sub-section 7.2.1 describing use cases and requirements of the external model enrichment process. The following sub-section 7.2.2 describes the development of the enrichment process and illustrates the process flow. Then sub-section 7.2.3 describes initial extraction strategies for adding triples to the external model.

7.2.1 USE CASES AND REQUIREMENTS

The external model enrichment process is proposed as an automated process. The implementation of the software prototype should not have any interaction with the user except running the software. To implement the enrichment process, the software should have the following functionalities:

- **Access to DOM data of a web page** – To extract data from the DOM, the software should be able to traverse into the DOM.
- **Extraction of semantic markup data** – The software should be able to identify semantic markup data on the web page and construct an RDF graph from it as the initial external model.
- **Extraction of additional triples from DOM data** – The software should be able to extract data from the DOM that would likely be relevant to the internal model. These new data is added to the external model.
- **Data export** – To use the enriched external model in mapping, the software should be able to serialise the model in a format that can be used in mapping tools. The format used in the previous experiment was RDF/XML.

The following sub-section describes an initial set of extraction strategies that can be used to extract additional properties from the DOM.

7.2.2 Overview of the External Model Enrichment Process

This section describes the development of the external model enrichment process. The flow of the external model enrichment process is illustrated in Figure 19. A web page is an input to the process. A DOM tree is created from the web page by a web rendering engine component such as can found in any web browser. Meanwhile, an initial external model is created using the semantic data markup parser component. Links between HTML elements and the semantic markup data that they are annotated with, are stored for future reference in the DOM-semantic markup data link table.

A set of extraction strategies then extract additional triples from the DOM. These additional triples will be added to the external model and result in the enriched external model. Links between HTML elements and the new extracted triples are also stored for future reference.

The next sub-section discusses the data that can be extracted from the DOM. An initial set of extraction strategies is also proposed.

7.2.3 Extraction Strategies

As discussed in section 7.1 above, the ultimate goal of external model enrichment is for the external model to have as much data that is relevant to the internal model as possible. Therefore, the choice of data being added to the external model depends on the internal model.

In the scope of this thesis, internal models are in the format of the Web Ontology Language or OWL. Thus, the data of internal models is divided into classes, properties and individuals – instances of either a class or a property. External model enrichment would then have to specify whether the added data is a class, a property or an individual.

Since this thesis focuses on web pages that describe single instances, extracting data from them is adding individuals to the internal model. In addition, with web pages describing single instances, it is unlikely for the web pages to contain data about additional classes that can be added to the external model. Therefore, the proposed external model enrichment process is focused on only adding properties.

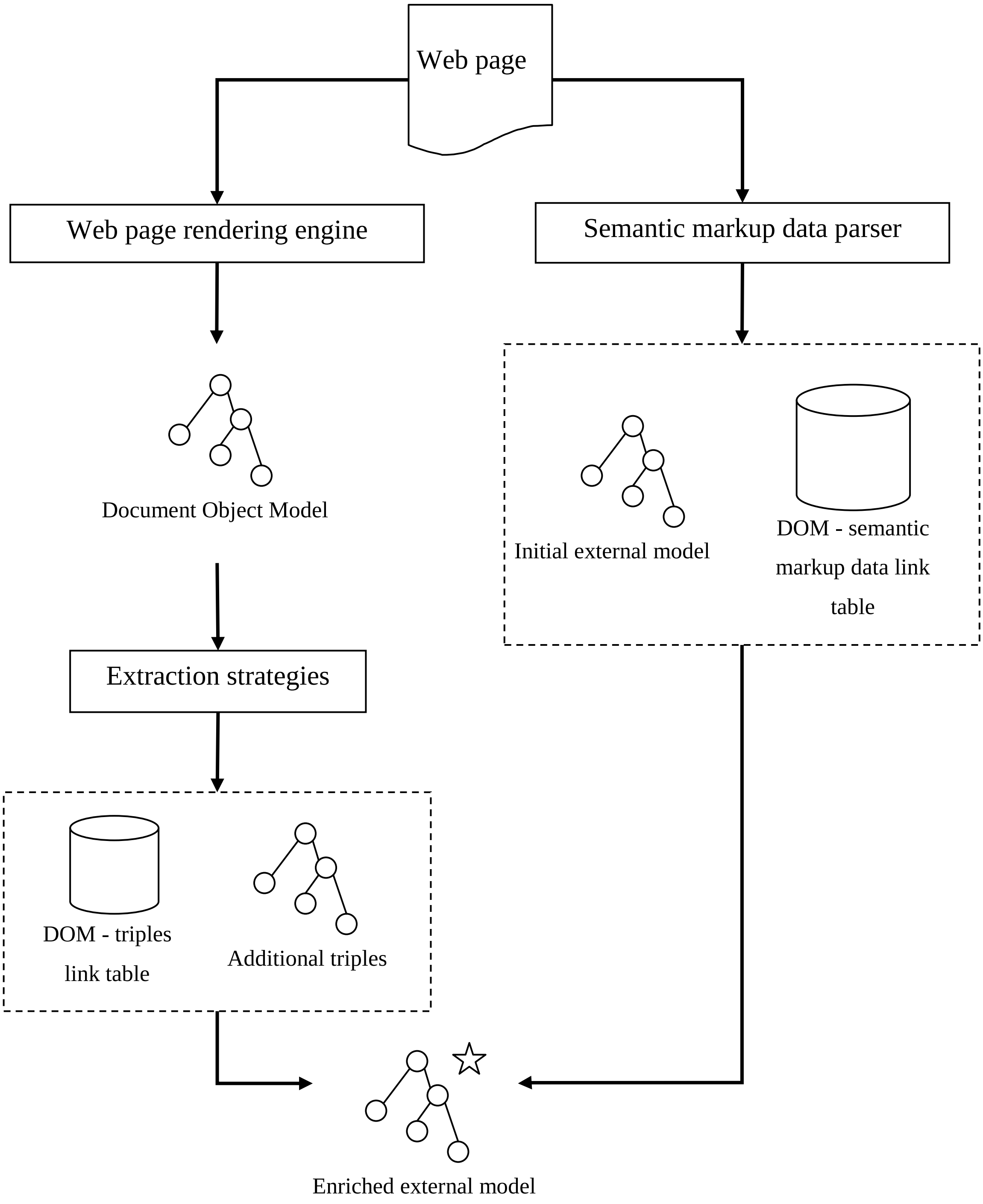


Figure 19: Flow of the External Model Enrichment Process

RDF properties denote relations between subjects and objects. In this case, subjects already exist and are identifiable with semantic markup data in the DOM. Semantic markup data already uses DOM data as objects in triples. This means additional properties will be using DOM data as objects as well. Property names usually signify the relations the properties represent. Therefore, it is unlikely that DOM data that is not in natural language would be suitable as new properties.

In order to find additional properties to add to the external model, patterns of data in the DOM that can be considered as properties have to be devised. By observing DOM data of 20 websites that contain most number of triples for each of Microformats, RDFa and Microdata (Mika and Potter 2012), the author of the thesis found that these patterns can be categorised into three levels: the attribute level, the element level and the structure level.

Based on the observed patterns, strategies for extracting data to be added as properties in the external model were created. The strategies for each level are discussed below.

Attribute Level – Strategy #1

As discussed in sub-section 2.2.1, HTML elements are assigned additional data through attributes. Attributes are also used for annotation with semantic markup data. HTML5 provides 15 global attributes[75]. From observation, there are three attributes whose value is arbitrary text that is usually in natural language. They are the attributes *class* and *id*.

According to the specification document[76], the purpose of *class* is to specify the classes to which an element belongs. These classes can be used arbitrarily by the web browser. Style sheet values can be assigned to an element based on its class. The specification document notes that: “authors are encouraged to use values that describe the nature of the content, rather than values that describe the desired presentation of the content.”

The attribute *id* is used to assign a unique identifier to an HTML element. The specification document suggest the attribute id is used “as a way to link to specific parts of a document using fragment identifiers, as a way to target an element when scripting, and as a way to style a specific element from CSS.”

[75] W3C, “HTML 5.1 Nightly: A vocabulary and associated APIs for HTML and XHTML” http://www.w3.org/html/wg/drafts/html/master/dom.html. Note that this number of attributes excludes event handlers such as *onclick*. Event handlers of an HTML element are attributes that are used to control triggers for JavaScript functions based on certain interactions the element is under. They are excluded by the author of thesis because, in the author’s opinion, they are used more for controlling the element’s presentation and rarely provide any meaning to the element.

[76] ibid.

The author of the thesis has observed cases of *class* and *id* values coexisting with Microdata properties in the same elements. In many of these cases, the *class* and *id* values are annotations for titles of fragments of the web page that are relevant to the properties. For example, the local directory website Yelp[77] use an *id* value “bizPhone” in an element that is annotated with the Schema.org property *telephone*. The book retailer Barnes & Noble[78] uses a *class* value “product-image” together with the Schema.org property *image*. Thus the author of the thesis believes that these class and id values could serve as candidates for properties in the external model.

An extraction strategy **Strategy#1** is created from using values of the attributes *class* and *id* as properties. Pseudocode for **Strategy#1** is described in Figure 20. The string values from the attributes may have to be *normalised* to ensure that it can be serialised into an RDF/XML file later. The normalisation process consists of:

1. Removing leading and trailing whitespaces,
2. Substituting whitespaces with underscores,
3. Substituting characters that are not alphanumeric or underscores with underscores,
4. Adding an underscore in front of the string in case it does not begin with a character or an underscore

Element Level – Strategy #2 and Strategy #3

As discussed in sub-section 2.4.1, most of HTML elements are used mainly for presentation purposes. In other words, they are used to control appearances of the data rather than to define relationship among the data. However, among all the HTML elements[79], there are some that are used to separate the content into fragments. The author of the thesis would argue that; when a web page is separated into fragments of a similar format, and each fragment has a title, the title serves the purpose of describing the relation that the fragment

[77] Yelp, http://www.yelp.com, Last visited: March 2013.

[78] Barnes & Noble, http://www.barnesandnoble.com, Last visited: March 2013.

[79] HTML elements, http://www.w3.org/TR/html-markup/elements.html, Last visited: March 2013.

has to the page as a whole. HTML elements that fall into this category are the heading elements: H1, H2, H3, H4, H5 and H6.

```
Extract triples from existing Microdata data into the external
model.
Locate an HTML element with the attributes itemtype and itemscope.
Let subject be the value of itemtype.
Locate HTML elements with the attribute itemprop.
For each HTML element with the attribute itemprop:
  Let object be the value of the object of the Microdata property.
  If the element has the attribute class:
    Let classes be value of the attribute class.
    For each space separated string from classes:
      Let property be the normalised value of the string.
      Add a triple (subject,property,object) to the external model.
    End for.
  End if.
  If the element has the attribute id:
    Let property be the normalised value of the attribute id.
    Add a triple (subject,property,object) to the external model.
  End if.
End for.
```

Figure 20: Pseudocode for Extracting Properties from the Attributes Class and ID

This prompts the creation of **Strategy#2**. To separate a web page into fragments, the author believes that the heading elements have to be in the same hierarchical level in the DOM. In this strategy, values of these heading elements are used as properties. Pseudocode for Strategy#2 is described in Figure 21.

```
Extract triples from existing Microdata data into the external
model.

Locate an HTML element with the attributes itemtype and itemscope.

Let subject be the value of itemtype.

Locate H1, H2, H3, H4, H5 and H6 elements within the scope of
subject.

For each heading element:

  Let property be the normalised value from the heading element.

  Let object be a blank string.

  Add a triple (subject,property,object) to the external model.

End for.
```

Figure 21: Pseudocode for Extracting Properties from Heading Elements

Apart from HTML elements that separate the whole web page into fragments, there are also elements that point to external resources. The HTML element list calls them *embedded content*[80]. This manner of having elements for embedded content corresponds to the naming convention of OWL properties. It is suggested that OWL properties that describe data to be belonging to a subject begin with “has”[81]. Since there is a definite list of HTML elements for embedded content, a definite list of properties that refer to the embedded content can be created.

To add properties related to embedded content, **Strategy#3** is created. This strategy locates HTML elements for embedded content within the scope of an element annotated with a subject. Properties *hasX* where X refers to the data type of the embedded content are created. Pseudocode for Strategy#3 is described in Figure 22.

[80] HTML elements organized by function, http://www.w3.org/TR/html-markup/elements-by-function.html, Last visited: March 2013.

[81] OWL 2 Web Ontology Language Primer, http://www.w3.org/TR/owl2-primer, Last visited: March 2013.

```
Extract triples from existing Microdata data into the external
model.

Locate an HTML element with the attributes itemtype and itemscope.

Let subject be the value of itemtype.

Locate img, audio and video elements within the scope of subject.

For each located element:

  Let tagName be the tag name of the element.

  Let property be “has” + tagName.

  Let object be the source URL of the element.

  Add a triple (subject,property,object) to the external model.

End for.
```

Figure 22: Pseudocode for Extracting Properties from Embedded Content Elements

Structure Level – Strategy #4

HTML elements on a web page are arranged in a tree data structure. Elements may be nested within others and form a hierarchy. This hierarchy is significant in annotating HTML elements with semantic markup data. An HTML element annotated with a property must be a descendant of an element annotated with a subject. This property may exist "deeper" than one level from the subject. In other words, there could be layers of HTML elements between the subject and the property elements. It was discussed previously above that some free text attributes may be used as properties. Such attributes may exist in these intermediary elements.

The author would contend that sometimes the hierarchical structure of HTML provides a sense of subclass between elements. If these nested elements contain property candidates like in **Strategy#1** and exist in multiple levels between a subject and a property, the property candidates should serve as property candidates to the parent subject. **Strategy#4** which uses this method finds all values from the attributes *class* and *id* between elements annotated with a subject and a property. The strategy then adds those values as property candidates. Pseudocode for Strategy#4 is described in Figure 23.

```
Extract triples from existing Microdata data into the external
model.

Locate an HTML element with the attributes itemtype and itemscope.

Let subject be the value of itemtype.

Locate HTML elements with the attribute itemprop.

For each HTML element with the attribute itemprop:

  Let betweenList be a list of elements in the hierarchy between the
  subject element and the current element.

  Let object be the value of the object of the Microdata property.

  For each HTML element in betweenList:

    If the element has the attribute class:

      Let classes be value of the attribute class.

      For each space separated string from classes:

        Let property be the normalised value of the string.

        Add a triple (subject,property,object) to the external
        model.

      End for.

    End if.

    If the element has the attribute id:

      Let property be the normalised value of the attribute id.

      Add a triple (subject,property,object) to the external model.

    End if.

  End for.

End for.
```

Figure 23: Pseudocode for Extracting Properties from the Attributes Class and ID between the Subject Element and Property Elements

7.3 IMPLEMENTATION

This section describes the implementation of the external model enrichment process as a proof-of-concept software prototype. The section begins with an overview of the software prototype in sub-section 7.3.1. The following sub-section 7.3.2 describes the implementation of the enrichment process in details.

7.3.1 OVERVIEW

The software prototype was implemented as a Google Chrome web browser extension. In this way the software prototype would have access to DOM data on web pages and some of the code from the previous experiment could be reused. Moreover, both software prototypes can be combined in the future without much change in terms of software architecture.

Figure 24 illustrates the architecture of the prototype software. Data in the web page consists of the DOM data and the embedded semantic markup data. The semantic markup data scattered on the web page is combined into an RDF graph by a semantic markup parser component. Additional triples are extracted from the DOM by the DOM data extractor. The DOM data extractor contains the four extraction strategies described in sub-section 7.2.3. The original external model from the semantic markup data and the additional triples from the DOM data extractor are then combined into the enriched external model in the triple store. This enriched external model can then be exported from the prototype for mapping.

7.3.2 THE EXTERNAL MODEL ENRICHMENT PROCESS

For DOM manipulation and Microdata parsing, the software prototype uses three JavaScript libraries: jQuery[82], rdfQuery[83] and MicrodataJS[84]. jQuery provides DOM traversal and manipulation functionalities such as element selection and attribute modification. rdfQuery provides a triple store and allows exporting an RDF graph in the RDF/XML format. jQuery and rdfQuery were also used in the previous software prototype.

[82] jQuery, http://jquery.com, Last visited: March 2013.

[83] rdfQuery, http://code.google.com/p/rdfquery, Last visited: March 2013.

[84] MicrodataJS, http://gitorious.org/microdatajs, Last visited: March 2013.

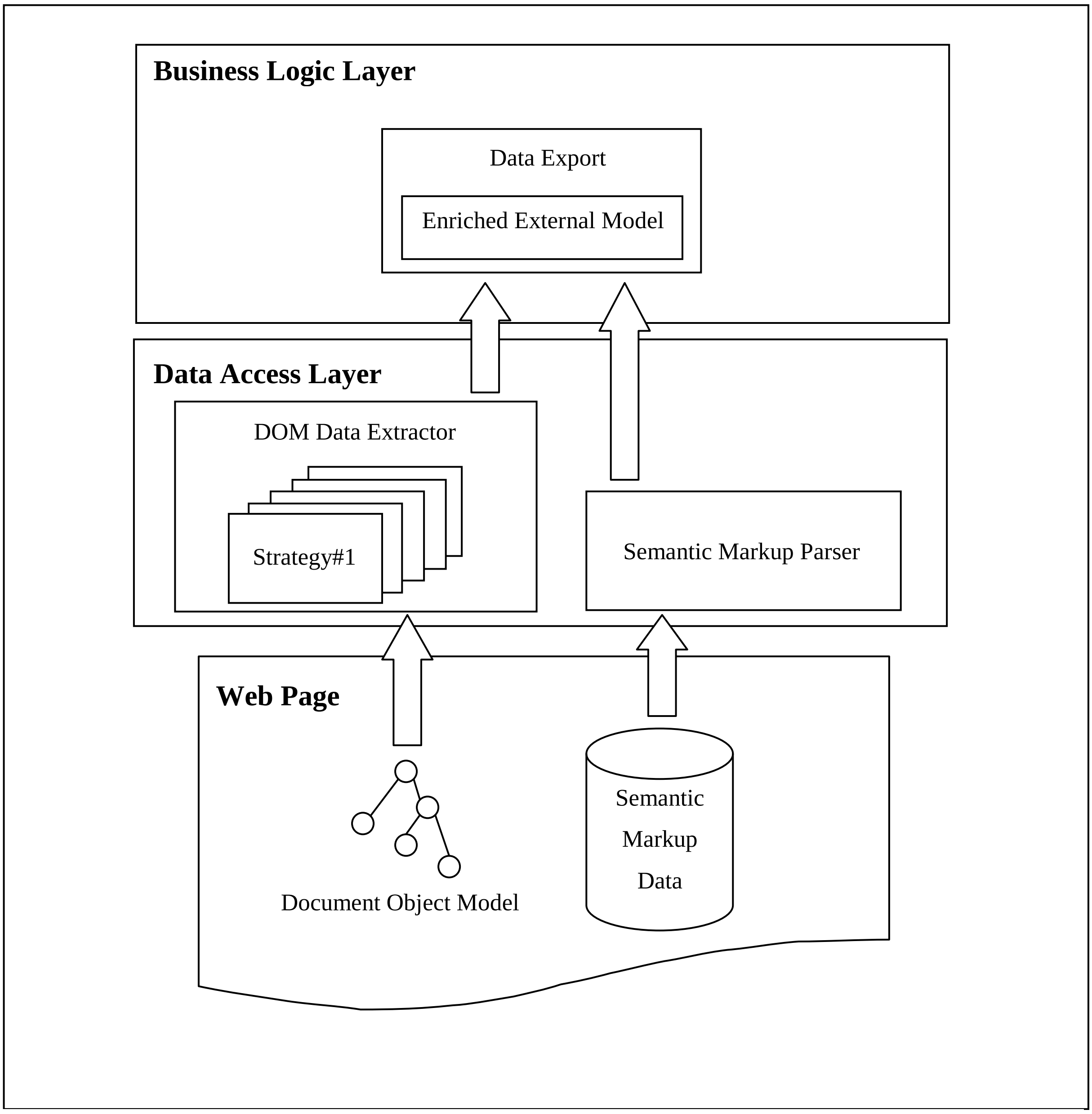


Figure 24: Architecture of the prototype software for the External Model Enrichment Process

MicrodataJS allows extracting Microdata as RDF from an HTML document. It was implemented according to a W3C specification document[85]. The MicrodataJS library can also be used to locate HTML elements that contain Microdata.

The strategies described in sub-section 7.2.3 were also implemented in JavaScript. The source code files can be found on the DVD accompanying this thesis.

[85] W3C, “Microdata to RDF: Transformation from HTML+Microdata to RDF”, http://www.w3.org/TR/microdata-rdf/, Last visited: March 2013

7.4 SUMMARY

This chapter describes a process of improving web data extraction called external model enrichment process. The motivation of developing the external model enrichment process comes from the fact that semantic markup data is likely not to be enough to annotate all the data on a web page. Thus using semantic markup data alone in mapping an external model to an internal model may not have the most coverage of the data available.

The enrichment process adds new properties to the external model. These properties are extracted from the document object model representation of the web page. Four extraction strategies were created as an initial set to explore the usefulness of the DOM as a resource for extra properties.

A software prototype implementation of the enrichment process is described as a web browser extension. This allows the prototype to have access to and manipulate the DOM and the semantic markup data.

In the next chapter, an experiment to evaluate the enrichment process is presented. The next chapter also analyses the experimental results and concludes with improvements and future work.

8 DOM-BASED EXTERNAL MODEL ENRICHMENT: EVALUATION

This chapter presents an evaluation of the external model enrichment process proposed in chapter 7. Section 8.1 outlines the objectives of the evaluation. Section 8.2 discusses the hypothesis. Section 8.3 describes how an experiment was designed for the evaluation. Section 8.4 describes the setup in which the experiment was conducted. Section 8.5 illustrates the results obtained from the experiment. Section 8.6 discusses an analysis of the results. The chapter concludes in section 8.7.

8.1 OBJECTIVES

According to the use case in chapter 4, web data extraction by human users is considered ideal. The proposed external model enrichment process is aimed at trying to reduce human effort while maintaining the quality of the results. This evaluation was conducted to:

- Measure the performance of the proposed enrichment process, and
- Identify possibilities for improvement of the enrichment process.

8.2 HYPOTHESIS

In this evaluation, the following hypothesis was used:

> *The four extraction strategies for identifying additional properties in DOM data and adding them to an initial external model help to improve the precision and recall of web data extraction.*

8.3 EXPERIMENT DESIGN

This section describes how an experiment was designed to test the hypothesis. To measure any improvements in web data extraction caused by the enrichment process, its results have to be compared to the results with the enrichment process “switched off”. In other words, two approaches of obtaining mapping correspondences, one with and one without the enrichment process, have to be compared. In the experiment, these two approaches are labelled the *baseline* approach and the *experimental* approach, respectively.

Moreover, both approaches have to be compared to the *gold standard* correspondences or the best available correspondences. These correspondences are created manually by the author as in this thesis human judgement is considered the most desirable.

First, sub-section 8.3.1 describes the baseline approach and the experimental approach. Then, sub-section 8.3.2 describes the gold standard results. Subsequently, sub-section 8.3.3 describes the overall experimental process. Finally, sub-section 8.3.4 describes the analysis method used to evaluate the results.

8.3.1 APPROACHES

In the external model enrichment process, additional data beyond just semantic markup data is used during the mapping to an internal model. Therefore, the performance of the enrichment process can be measured against the original process by analysing the results of when the additional data is used compared with when the additional data is not used. In the experiment, two approaches to obtaining mapping correspondences are used: the baseline approach and the experimental approach.

The Baseline Approach: This approach uses an external model comprising only semantic markup data of the web page. The external model is mapped to an internal model with the Alignment API, a semantic mapping tool.

The Experimental Approach: This approach uses an external model that is enriched with added properties from the DOM. The external model in this approach is also mapped to the internal model using the Alignment API.

8.3.2 GOLD STANDARD RESULTS

As described in chapter 4, the ideal situation of web data extraction is where the human user indicates which part of data on the web page should be mapped to which term in the internal model. The data on the web page is usually represented as a DOM tree that offers unique paths to each element. Thus, the gold standard correspondences in this case were chosen by the author as matches between DOM paths of HTML elements and terms in an internal model.

This definition of gold standard correspondences means the results have a different format from the correspondences generated by the baseline approach and the experimental approach, which are in the Alignment Format. This issue is addressed by having a DOM-triples link table shown in Figure 19. The DOM-triples link table links maps terms in the external model to the originating HTML elements. Thus it connects the mapped terms in the internal model to the HTML elements.

8.3.3 Overall Experimental Process

The experimental process is illustrated in Figure 25. The process can be described as the following steps:

1. Locate the target web page.
2. Save the target web page into an HTML file.
3. Prepare the web page for using in a dataset.
    - Add protocols to protocol-less URLs.
    - Remove other subjects that do not use the properties on the page.
    - Add “www” to URIs without “www”.
    - Remove JavaScript code from the web page.
4. Construct external models from both the baseline approach and the experimental approach.
5. Export the external models into files.
6. Generate mapping correspondences between the external model and an internal model.
7. Identify origin HTML elements of mapped terms.

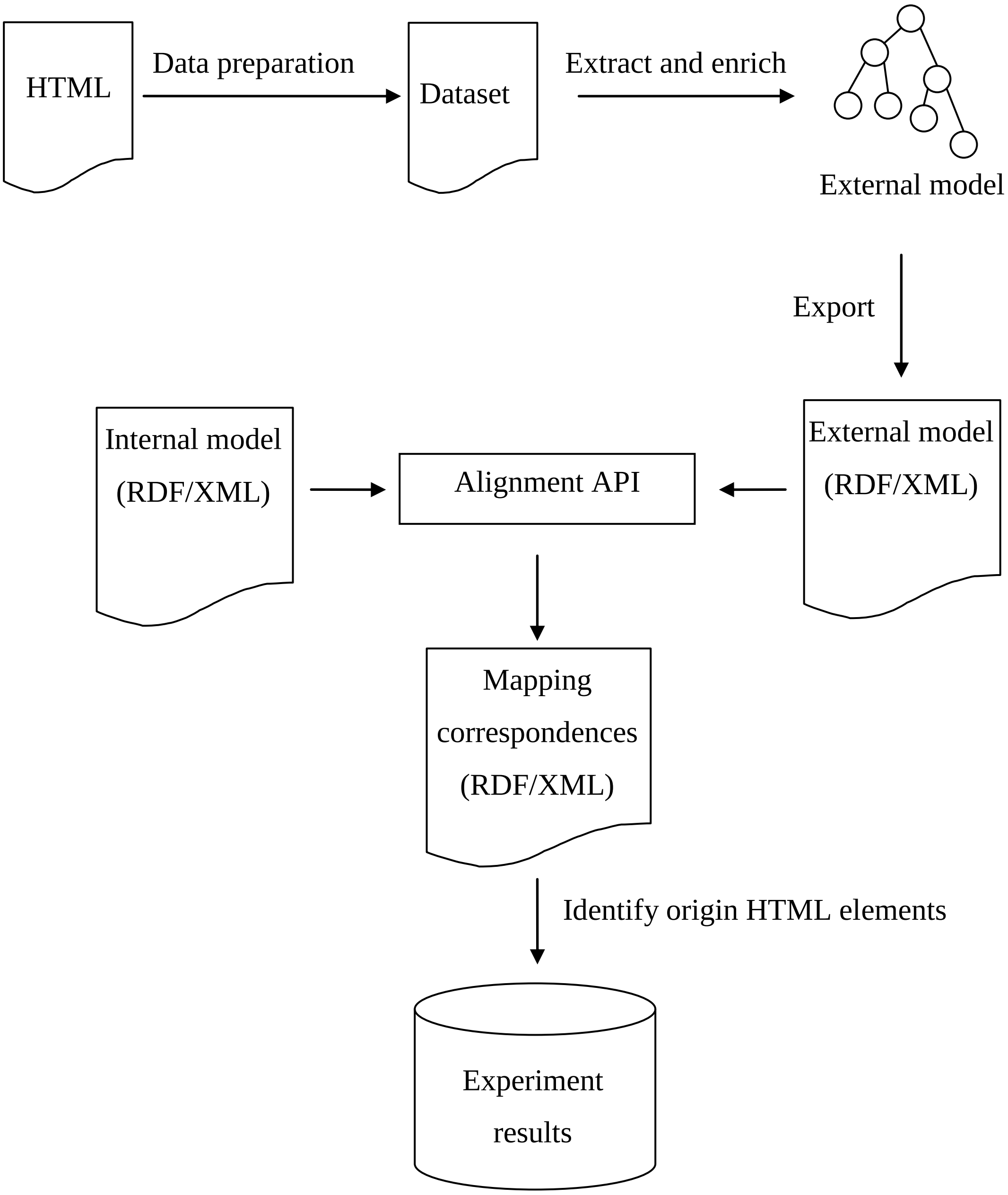


Figure 25: Experimental Process

8.3.4 *Analysis Methodology*

Correspondences from both the baseline approach and the experimental approach are to be compared against the gold standard correspondences. For each approach, there would be HTML elements that are included and not included in the external model, either by the publisher of the semantic markup data or by the extraction strategies. The number of these HTML elements and the number of terms in the internal model can be used to determine the number of possible matches by a simple combinatorial concept.

The concept is: if there are M HTML elements and N terms in the internal model available for matching, there will be M × N possible matches. Let us suppose that there are X possible matches. Four types of outcome can be expected when comparing the baseline approach and the experimental approach to the gold standard. These outcomes are labelled as described below.

- **True Positive (TP)**: These are the matches that exist in the gold standard and also occur in the generated correspondences of the approaches.
- **True Negative (TN)**: These are the matches that do not exist in the gold standard and also do not occur in the generated correspondences of the approaches.
- **False Positive (FP)**: These are the matches that do not exist in the gold standard but occur in the generated correspondences of the approaches.
- **False Negative (FN)**: These are the matches that exist in the gold standard but do not occur in the generated correspondences of the approaches.

To measure accuracy, one can use the ratio between correct results (true positive and true negative) and the total results. This can be written as shown in Equation 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 1: Accuracy

However, some problem may arise due to the large amount of HTML elements on a web page. It has been shown that; as of late 2012, the average number of HTML elements on a web page is about 1,000[86]. The number of terms in an internal model could be much less than 1,000. For example, in the Movie Ontology used in this experiment, the total number of class and property terms altogether is 120.

An illusion of high accuracy may happen if the number of true negative cases is relatively larger than that of the false negative cases. Figure 26 illustrates this situation. In the example, the number of HTML elements on the web page is 1,000. The number of terms in the internal model is 100. This makes the total number of possible matches to be 1000 × 100 = 100,000 matches. Let us suppose that all 100 terms in the internal model have their matches in the gold standard. However, let us also suppose that the web data extraction process does not generate any results at all. This means the web data extraction process still has a relatively large number of true negative results and achieves the accuracy of 99.90%. This seemingly high accuracy does not reflect the fact that the web data extraction process does not yield any results.

Number of HTML elements:	1,000
Number of terms in the internal model:	100
Number of possible matches:	100,000

	Exist in Results	Do Not Exist in Results
Exist in Gold Standard	0	100
Do Not Exist in Gold Standard	0	99,900

$$Accuracy = \frac{0+99{,}900}{0+99{,}900+0+100} = 99.9\%$$

Figure 26: Accuracy Paradox

[86] The HTTP Archive, http://www.httparchive.org, Last visited: March 2013.

To avoid this paradox, other metrics are used in the evaluation. Instead of incorporating the unavoidably large number of true negative results, one may look at only the results that the web data extraction process yields. By finding the ratio between the correct matches and all the matches the web data extraction process yields, one may have a sense of how precise the process is. In literature, this metric is called *precision*. It is defined as shown in Equation 2.

$$Precision = \frac{TP}{TP + FP}$$

Equation 2: Precision

Let us reuse the example in Figure 26. The precision in this case is 0%. If the web data extraction process in this situation instead yielded 50 matches, 10 of which were correct, it would achieve the accuracy and precision of 99.91% and 20%, respectively.

The precision metric however does not offer any information on the number of matches generated by the web data extraction process. In the example from Figure 26, if the process yielded only one match which happened to be correct, it would achieve the precision of 100%. This seems desirable despite the fact that many correct matches were ignored. In this case, the ratio between the correct matches and the ideal matches can provide the sense of how much relevant data was actually generated. In literature, this metric is called *recall*. It is defined as shown in Equation 3.

$$Recall = \frac{TP}{TP + FN}$$

Equation 3: Recall

In the case of having only one correct matches out of the desirable 100, the web data extraction process achieved the recall of only 1%.

In literature, precision and recall are often combined together as one metric called the F-measure score and is defined as shown in Equation 4.

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Equation 4: F-measure

The F-measure score is a harmonic mean of precision and recall[87]. It has been used in both the field of information extraction (Makhoul et al. 1999) and semantic mapping (Euzenat and Shvaiko 2007).

In summary, there are two approaches to obtaining correspondences in this experiment. In the first approach – the baseline approach – web data extraction correspondences are obtained without using the external model enrichment process. In the other approach – the experimental approach – the enrichment process is used. Correspondences generated from both approaches are compared to the gold standard represents the ideal results created by a human user. Metrics used in the evaluation are precision, recall and the F-measure score.

8.4 Experiment Setup

This section describes the setup of the experiment. The section begins with describing the environment in which the experiment was conducted. The section then proceeds to discuss the datasets used in the experiment.

8.4.1 Environment

The whole experiment was conducted on a Dell laptop computer with 2.53GHz CPU and 4GB RAM. The experiment was implemented using the software tool, which is a web browser extension for Google Chrome.

For semantic mapping, Alignment API version 3.6 was used. The newer versions of Alignment API could not be used because they are stricter to mapping between OWL ontologies and expect OWL-specific properties while external models are usually only RDF.

8.4.2 Datasets

To ensure that the external model enrichment process can be generalised to different web pages, the experiment is conducted from data from three domains: films, literature and music. These domains were chosen by the author of the thesis to represent web data found in everyday use. For each domain, external models were extracted from two web pages from two websites. Only the data from the films domain is shown in details. The rest of the dataset can be found in the DVD accompanying this thesis.

[87] F-measure uses the harmonic mean because it is more suitable to ratios than the arithmetic mean.

Films

The external models for the film domain were taken from the Internet Movie Database (IMDb)[88] and the film review website Rotten Tomatoes (RT)[89]. They both use the Schema.org vocabulary's Movie data type definition.

The internal model was taken from the Movie Ontology project[90] which is maintained by the University of Zurich. It is an OWL graph with OWL-DL features such as inverse properties. This ontology is published and maintained by the University of Zurich.

Statistical figures of each model are described below.

IMDb

For the external model from IMDb, two classes and 16 properties are taken from the Schema.org vocabulary. Table 7 describes this model.

Rotten Tomatoes

For the external model from Rotten Tomatoes, three classes and 16 properties are taken from the Schema.org vocabulary. Table 8 describes this model.

The Movie Ontology

The Movie Ontology contains 78 classes, 42 properties, 17 of which are inverse properties, and 281 individuals. The full list of concepts represented in the Movie Ontology can be found at the Movie Ontology project website[91].

[88] The Internet Movie Database, http://www.imdb.com, Last visited: March 2013.

[89] Rotten Tomatoes, http://www.rottentomatoes.com, Last visited: March 2013.

[90] The Movie Ontology, http://www.movieontology.org, Last visited: March 2013.

[91] Ibid.

Table 7: Schema.org Classes and properties used in the IMDb model

Classes	Object properties	Data properties
AggregateRating Movie	actors aggregateRating director genre image inLanguage trailer	bestRating contentRating datePublished description duration name ratingCount ratingValue reviewCount

Table 8: Schema.org Classes and properties used in the Rotten Tomatoes model

Classes	Object properties	Data properties
AggregateRating Movie Person	actors aggregateRating director image url	awards bestRating contentRating datePublished description duration name productionCompany ratingValue reviewCount worstRating

Literature

The external models for the literature domain were taken from the book retailer Barnes & Noble[92] and the book review website Goodreads[93]. Barnes & Noble uses the Schema.org vocabulary's Product data type definition and Goodreads uses the Schema.org vocabulary's Book data type definition.

The internal model for this domain was taken from a metadata namespace of the publishing company O'Reilly Media. The O'Reilly metadata namespace is an OWL Full graph.

Statistical figures of each model are described below.

[92] Barnes & Noble, http://www.barnesandnoble.com/, Last visited: April 2013.

[93] Goodreads, http://www.goodreads.com, Last visited: April 2013.

Barnes & Noble

For the external model from Barnes & Noble, five classes and 12 properties are taken from the Schema.org vocabulary.

Goodreads

For the external model from Goodreads, four classes and 10 properties are taken from the Schema.org vocabulary.

O'Reilly

The O'Reilly metadata namespace contains 16 classes, 73 properties, and 7 individuals. The full list of concepts represented in the O'Reilly metadata namespace can be found at the namespace's URI[94].

Music

The external models for the music domain were taken from the online radio website Last.fm[95] and the music-oriented social network MySpace[96]. Both models use the Schema.org's MusicGroup and MusicRecording data type definitions.

The internal model for this domain was taken from the Music Ontology. The Music Ontology is an OWL Full graph.

Statistical figures of each model are described below.

Last.fm

For the external model from Last.fm, three classes and 7 properties are taken from the Schema.org vocabulary.

[94] O'Reilly Metadata Namespace, http://purl.oreilly.com/ns/meta/, Last visited: April 2013.

[95] Last.fm, http://last.fm, Last visited: April 2013.

[96] MySpace, http://www.myspace.com, Last visited: April 2013.

MySpace

For the external model from MySpace, two classes and five properties are taken from the Schema.org vocabulary.

The Music Ontology

The Music Ontology contains 79 classes, 182 properties, and 15 individuals. The full list of concepts represented in the Music Ontology can be found at the Music Ontology website[97].

8.5 RESULTS

For each web page, two sets of mapping correspondences were generated. One was from the baseline approach and another from the experimental approach. The results are discussed in detail for one data set. The rest are discussed only in terms of figures in this thesis. This is due to the amount of data involved. The full result set can be found in both RDF and Comma-Separated Value (CSV) file formats in a DVD accompanying this thesis.

The results from the IMDb dataset are shown below.

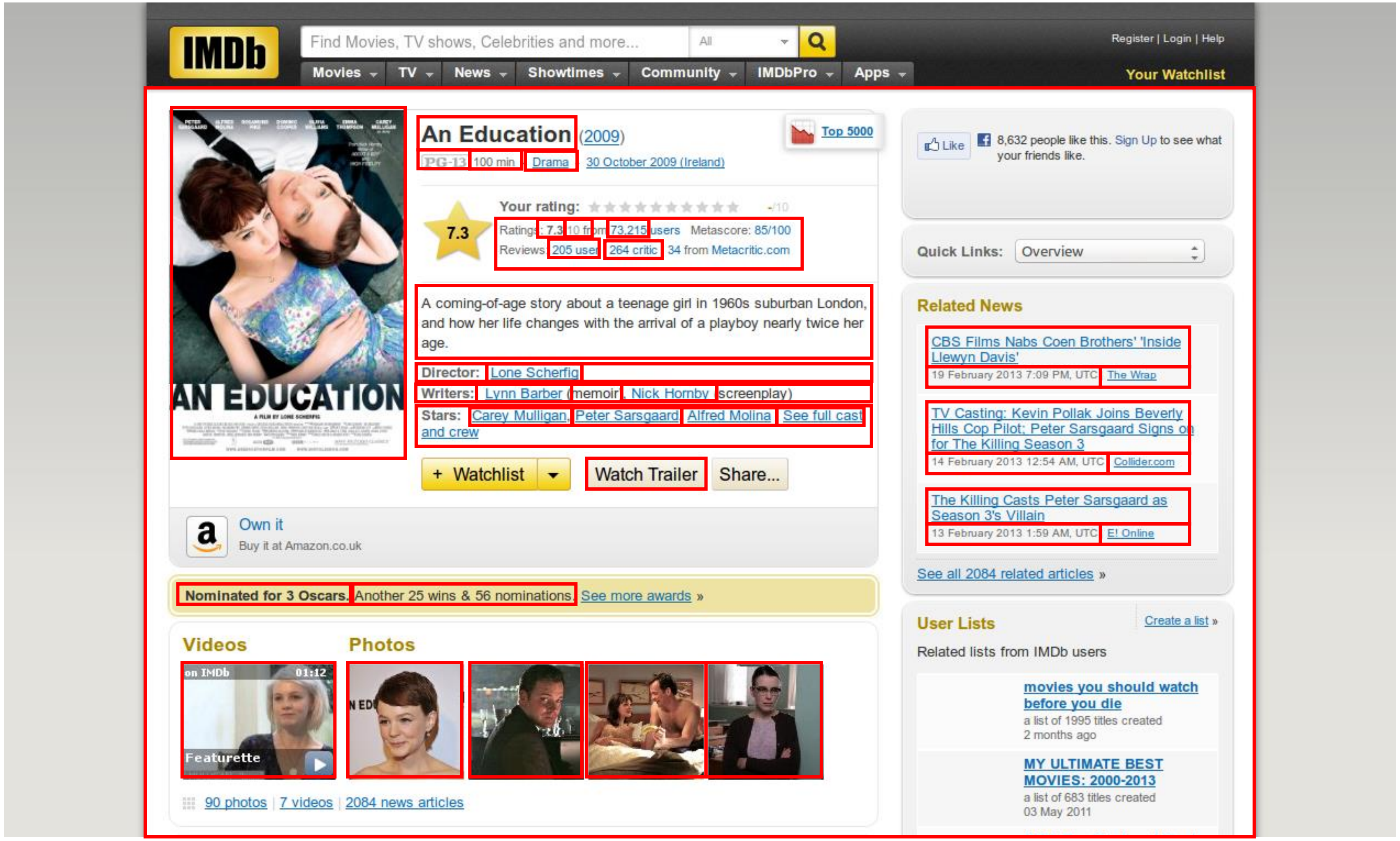
8.5.1 IMDB - THE BASELINE APPROACH

Figure 27 illustrates the HTML elements that are annotated with semantic markup data. These elements can be referred to with their unique DOM paths. A DOM path contains the path that leads from the web page's root element, the HTML element, to the target element. For example, a DOM path of an image may be written as follows:

```
html > body > div[2] > img[0]
```

The example DOM path above would lead to the first image element in the second div element in the body of the web page. The elements' corresponding semantic markup data and the mapped term from the internal model are shown together in Table 9.

[97] The Music Ontology, http://musicontology.com/, Last visited: April 2013.



An Education (2009)

PG-13 100 min Drama 30 October 2009 (Ireland)

7.3

Your rating: -/10

Ratings: 7.3/10 from 73,215 users Metascore: 85/100

Reviews: 205 user 264 critic 34 from Metacritic.com

A coming-of-age story about a teenage girl in 1960s suburban London, and how her life changes with the arrival of a playboy nearly twice her age.

Director: Lone Scherfig

Writers: Lynn Barber (memoir), Nick Hornby (screenplay)

Stars: Carey Mulligan, Peter Sarsgaard, Alfred Molina | See full cast and crew

Own it
Buy it at Amazon.co.uk

Nominated for 3 Oscars. Another 25 wins & 56 nominations. See more awards »

Videos

Photos

90 photos | 7 videos | 2084 news articles

8,632 people like this. Sign Up to see what your friends like.

Quick Links: Overview

Related News

CBS Films Nabs Coen Brothers' 'Inside Llewyn Davis'
19 February 2013 7:09 PM, UTC | The Wrap

TV Casting: Kevin Pollak Joins Beverly Hills Cop Pilot; Peter Sarsgaard Signs on for The Killing Season 3
14 February 2013 12:54 AM, UTC | Collider.com

The Killing Casts Peter Sarsgaard as Season 3's Villain
13 February 2013 1:59 AM, UTC | E! Online

See all 2084 related articles »

User Lists

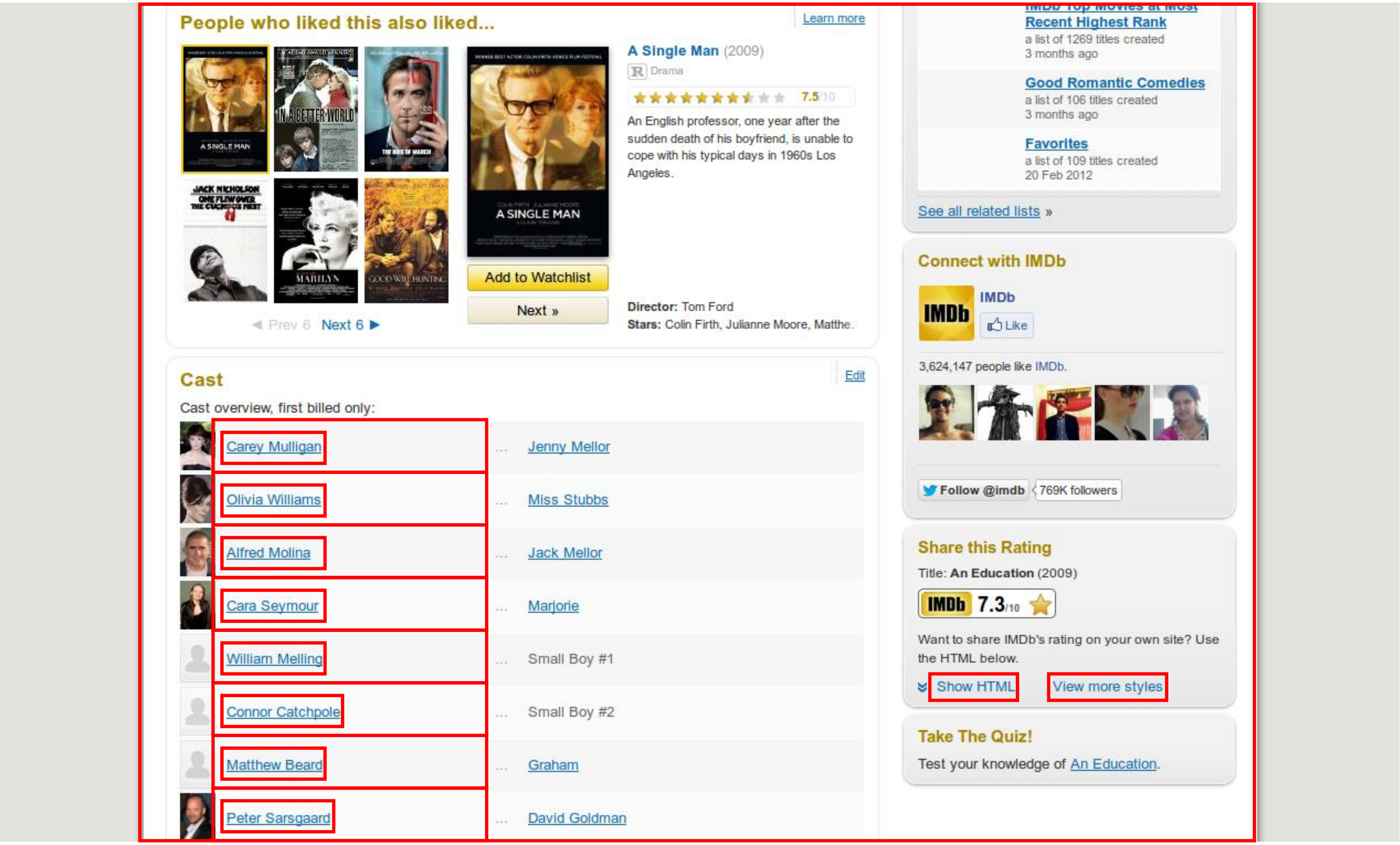
Create a list »

Related lists from IMDb users

movies you should watch before you die
a list of 1995 titles created 2 months ago

MY ULTIMATE BEST MOVIES: 2000-2013
a list of 683 titles created 03 May 2011

Figure 27: HTML Elements Annotated by Semantic Markup Data



People who liked this also liked...

Learn more

A Single Man (2009)

R Drama

7.5/10

An English professor, one year after the sudden death of his boyfriend, is unable to cope with his typical days in 1960s Los Angeles.

Add to Watchlist

Next »

◀ Prev 6 Next 6 ▶

Director: Tom Ford

Stars: Colin Firth, Julianne Moore, Matthe...

Cast

Edit

Cast overview, first billed only:

Carey Mulligan	...	Jenny Mellor
Olivia Williams	...	Miss Stubbs
Alfred Molina	...	Jack Mellor
Cara Seymour	...	Marjorie
William Melling	...	Small Boy #1
Connor Catchpole	...	Small Boy #2
Matthew Beard	...	Graham
Peter Sarsgaard	...	David Goldman

IMDb Top Movies at Most Recent Highest Rank
a list of 1269 titles created
3 months ago

Good Romantic Comedies
a list of 106 titles created
3 months ago

Favorites
a list of 109 titles created
20 Feb 2012

See all related lists »

Connect with IMDb

IMDb

Like

3,624,147 people like IMDb.

Follow @imdb 769K followers

Share this Rating

Title: **An Education** (2009)

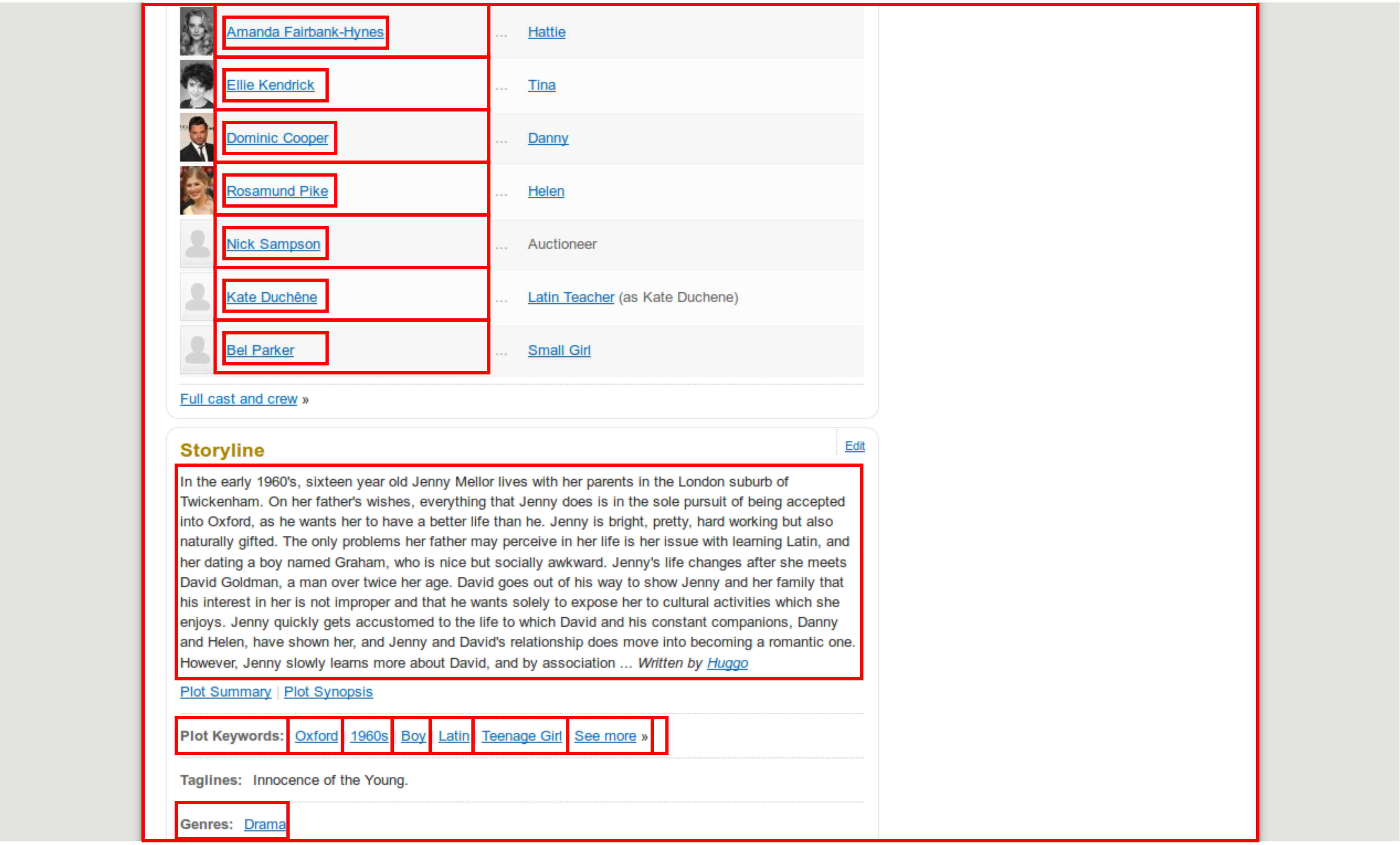
IMDb 7.3/10

Want to share IMDb's rating on your own site? Use the HTML below.

Show HTML View more styles

Take The Quiz!

Test your knowledge of An Education.



Amanda Fairbank-Hynes	...	Hattie
Ellie Kendrick	...	Tina
Dominic Cooper	...	Danny
Rosamund Pike	...	Helen
Nick Sampson	...	Auctioneer
Kate Duchêne	...	Latin Teacher (as Kate Duchene)
Bel Parker	...	Small Girl

Full cast and crew »

Storyline

Edit

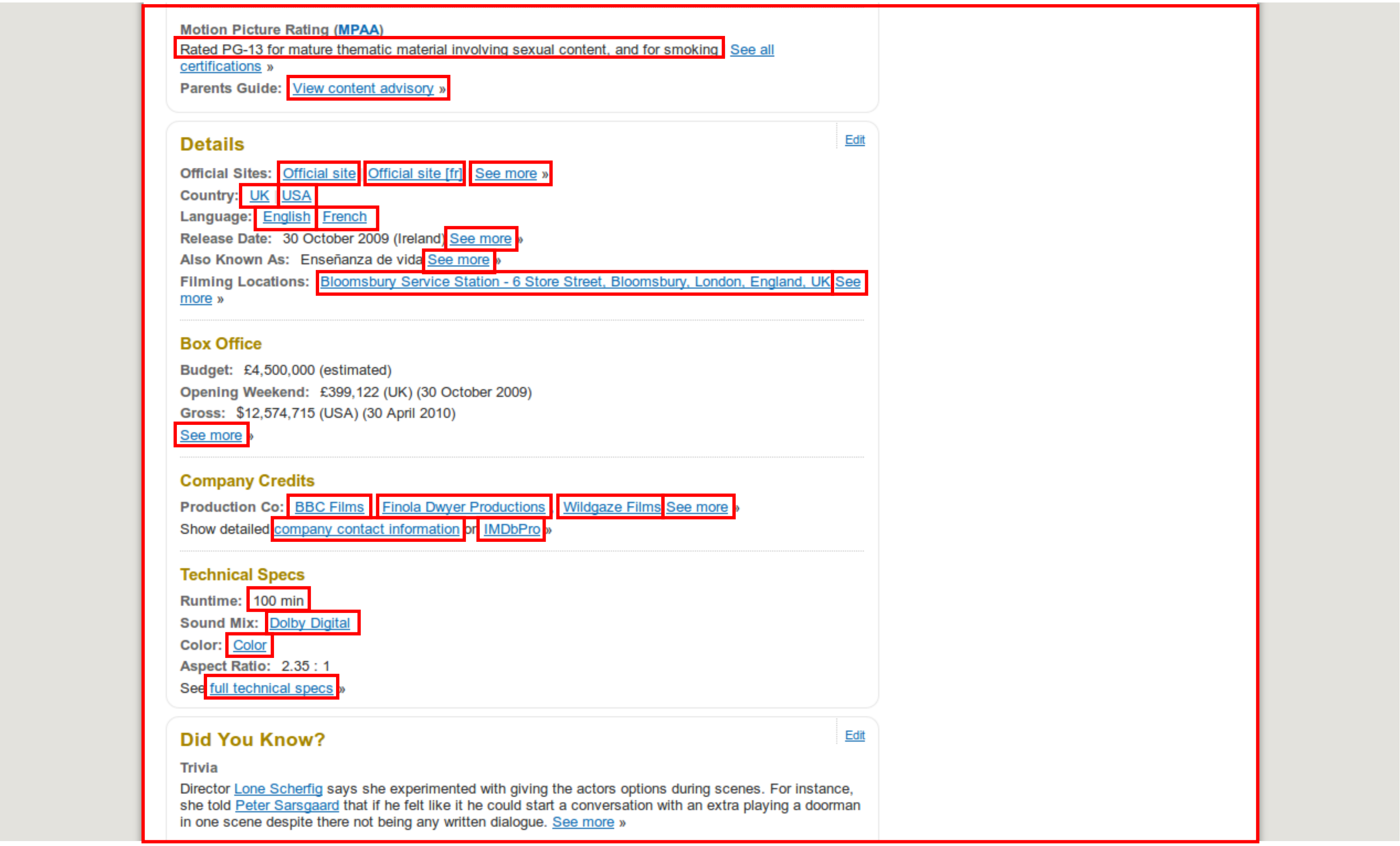
In the early 1960's, sixteen year old Jenny Mellor lives with her parents in the London suburb of Twickenham. On her father's wishes, everything that Jenny does is in the sole pursuit of being accepted into Oxford, as he wants her to have a better life than he. Jenny is bright, pretty, hard working but also naturally gifted. The only problems her father may perceive in her life is her issue with learning Latin, and her dating a boy named Graham, who is nice but socially awkward. Jenny's life changes after she meets David Goldman, a man over twice her age. David goes out of his way to show Jenny and her family that his interest in her is not improper and that he wants solely to expose her to cultural activities which she enjoys. Jenny quickly gets accustomed to the life to which David and his constant companions, Danny and Helen, have shown her, and Jenny and David's relationship does move into becoming a romantic one. However, Jenny slowly learns more about David, and by association ... *Written by Huggo*

Plot Summary | Plot Synopsis

Plot Keywords: Oxford | 1960s | Boy | Latin | Teenage Girl | See more »

Taglines: Innocence of the Young.

Genres: Drama



Motion Picture Rating (MPAA)

Rated PG-13 for mature thematic material involving sexual content, and for smoking See all certifications »

Parents Guide: View content advisory »

Details

Official Sites: Official site | Official site [fr] | See more »

Country: UK | USA

Language: English | French

Release Date: 30 October 2009 (Ireland) See more »

Also Known As: Enseñanza de vida See more »

Filming Locations: Bloomsbury Service Station - 6 Store Street, Bloomsbury, London, England, UK See more »

Box Office

Budget: £4,500,000 (estimated)

Opening Weekend: £399,122 (UK) (30 October 2009)

Gross: $12,574,715 (USA) (30 April 2010)

See more »

Company Credits

Production Co: BBC Films, Finola Dwyer Productions, Wildgaze Films See more »

Show detailed company contact information on IMDbPro »

Technical Specs

Runtime: 100 min

Sound Mix: Dolby Digital

Color: Color

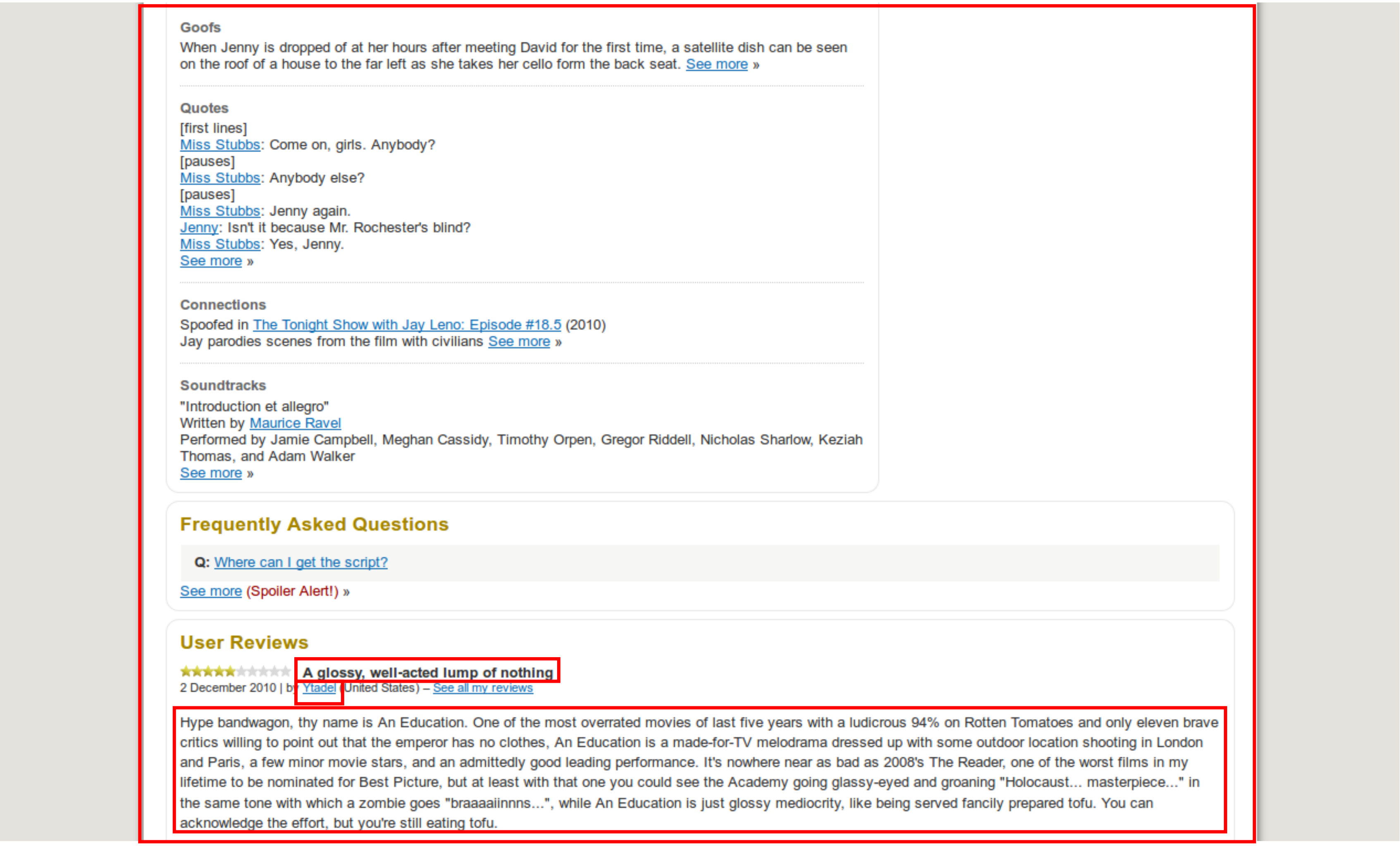
Aspect Ratio: 2.35 : 1

See full technical specs »

Did You Know?

Trivia

Director Lone Scherfig says she experimented with giving the actors options during scenes. For instance, she told Peter Sarsgaard that if he felt like it he could start a conversation with an extra playing a doorman in one scene despite there not being any written dialogue. See more »



Goofs

When Jenny is dropped of at her hours after meeting David for the first time, a satellite dish can be seen on the roof of a house to the far left as she takes her cello form the back seat. See more »

Quotes

[first lines]
Miss Stubbs: Come on, girls. Anybody?
[pauses]
Miss Stubbs: Anybody else?
[pauses]
Miss Stubbs: Jenny again.
Jenny: Isn't it because Mr. Rochester's blind?
Miss Stubbs: Yes, Jenny.
See more »

Connections

Spoofed in The Tonight Show with Jay Leno: Episode #18.5 (2010)
Jay parodies scenes from the film with civilians See more »

Soundtracks

"Introduction et allegro"
Written by Maurice Ravel
Performed by Jamie Campbell, Meghan Cassidy, Timothy Orpen, Gregor Riddell, Nicholas Sharlow, Keziah Thomas, and Adam Walker
See more »

Frequently Asked Questions

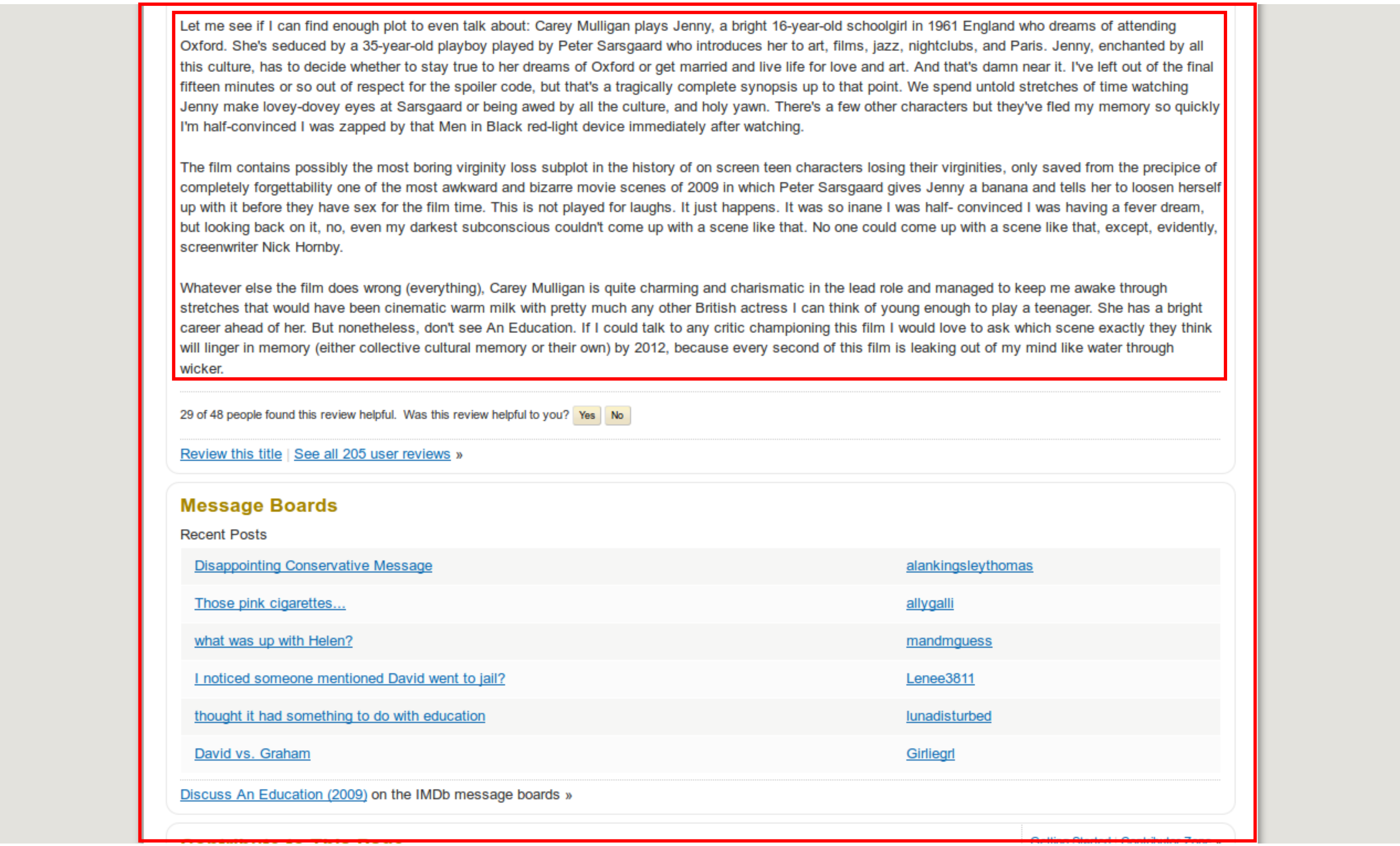
Q: Where can I get the script?

See more (Spoiler Alert!) »

User Reviews

★★★★★☆☆☆☆☆ **A glossy, well-acted lump of nothing**
2 December 2010 | by Ytadel (United States) – See all my reviews

Hype bandwagon, thy name is An Education. One of the most overrated movies of last five years with a ludicrous 94% on Rotten Tomatoes and only eleven brave critics willing to point out that the emperor has no clothes, An Education is a made-for-TV melodrama dressed up with some outdoor location shooting in London and Paris, a few minor movie stars, and an admittedly good leading performance. It's nowhere near as bad as 2008's The Reader, one of the worst films in my lifetime to be nominated for Best Picture, but at least with that one you could see the Academy going glassy-eyed and groaning "Holocaust… masterpiece…" in the same tone with which a zombie goes "braaaaiinnns…", while An Education is just glossy mediocrity, like being served fancily prepared tofu. You can acknowledge the effort, but you're still eating tofu.



Let me see if I can find enough plot to even talk about: Carey Mulligan plays Jenny, a bright 16-year-old schoolgirl in 1961 England who dreams of attending Oxford. She's seduced by a 35-year-old playboy played by Peter Sarsgaard who introduces her to art, films, jazz, nightclubs, and Paris. Jenny, enchanted by all this culture, has to decide whether to stay true to her dreams of Oxford or get married and live life for love and art. And that's damn near it. I've left out of the final fifteen minutes or so out of respect for the spoiler code, but that's a tragically complete synopsis up to that point. We spend untold stretches of time watching Jenny make lovey-dovey eyes at Sarsgaard or being awed by all the culture, and holy yawn. There's a few other characters but they've fled my memory so quickly I'm half-convinced I was zapped by that Men in Black red-light device immediately after watching.

The film contains possibly the most boring virginity loss subplot in the history of on screen teen characters losing their virginities, only saved from the precipice of completely forgettability one of the most awkward and bizarre movie scenes of 2009 in which Peter Sarsgaard gives Jenny a banana and tells her to loosen herself up with it before they have sex for the film time. This is not played for laughs. It just happens. It was so inane I was half- convinced I was having a fever dream, but looking back on it, no, even my darkest subconscious couldn't come up with a scene like that. No one could come up with a scene like that, except, evidently, screenwriter Nick Hornby.

Whatever else the film does wrong (everything), Carey Mulligan is quite charming and charismatic in the lead role and managed to keep me awake through stretches that would have been cinematic warm milk with pretty much any other British actress I can think of young enough to play a teenager. She has a bright career ahead of her. But nonetheless, don't see An Education. If I could talk to any critic championing this film I would love to ask which scene exactly they think will linger in memory (either collective cultural memory or their own) by 2012, because every second of this film is leaking out of my mind like water through wicker.

29 of 48 people found this review helpful. Was this review helpful to you?

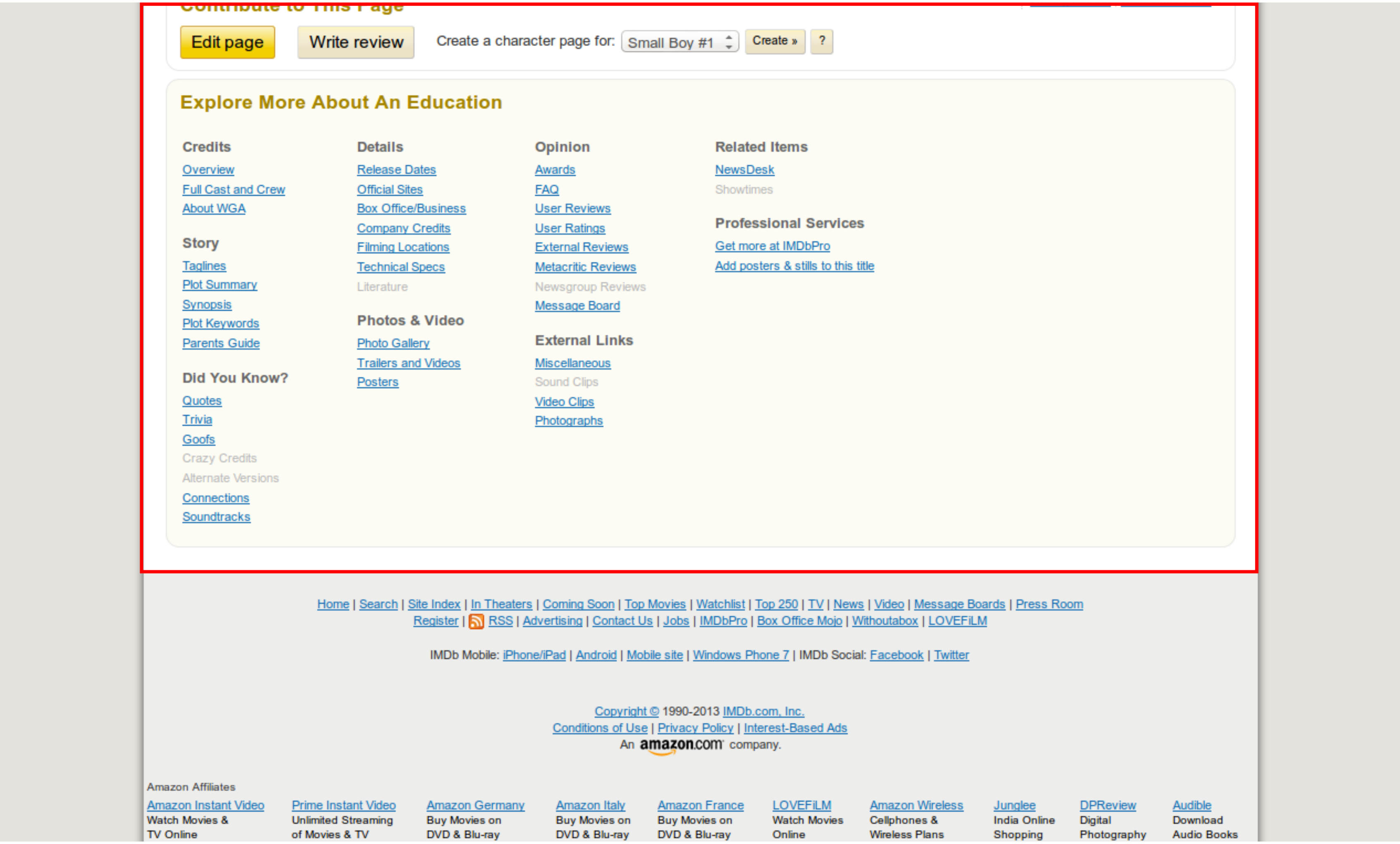
Review this title | See all 205 user reviews »

Message Boards

Recent Posts

Disappointing Conservative Message	alankingsleythomas
Those pink cigarettes...	allygalli
what was up with Helen?	mandmguess
I noticed someone mentioned David went to jail?	Lenee3811
thought it had something to do with education	lunadisturbed
David vs. Graham	Girliegrl

Discuss An Education (2009) on the IMDb message boards »



Contribute to This Page

Edit page | Write review | Create a character page for: Small Boy #1 | Create » | ?

Explore More About An Education

Credits
- Overview
- Full Cast and Crew
- About WGA

Story
- Taglines
- Plot Summary
- Synopsis
- Plot Keywords
- Parents Guide

Did You Know?
- Quotes
- Trivia
- Goofs
- Crazy Credits
- Alternate Versions
- Connections
- Soundtracks

Details
- Release Dates
- Official Sites
- Box Office/Business
- Company Credits
- Filming Locations
- Technical Specs
- Literature

Photos & Video
- Photo Gallery
- Trailers and Videos
- Posters

Opinion
- Awards
- FAQ
- User Reviews
- User Ratings
- External Reviews
- Metacritic Reviews
- Newsgroup Reviews
- Message Board

External Links
- Miscellaneous
- Sound Clips
- Video Clips
- Photographs

Related Items
- NewsDesk
- Showtimes

Professional Services
- Get more at IMDbPro
- Add posters & stills to this title

Home | Search | Site Index | In Theaters | Coming Soon | Top Movies | Watchlist | Top 250 | TV | News | Video | Message Boards | Press Room
Register | RSS | Advertising | Contact Us | Jobs | IMDbPro | Box Office Mojo | Withoutabox | LOVEFiLM

IMDb Mobile: iPhone/iPad | Android | Mobile site | Windows Phone 7 | IMDb Social: Facebook | Twitter

Copyright © 1990-2013 IMDb.com, Inc.
Conditions of Use | Privacy Policy | Interest-Based Ads
An amazon.com company.

Amazon Affiliates

Amazon Instant Video Watch Movies & TV Online	Prime Instant Video Unlimited Streaming of Movies & TV	Amazon Germany Buy Movies on DVD & Blu-ray	Amazon Italy Buy Movies on DVD & Blu-ray	Amazon France Buy Movies on DVD & Blu-ray	LOVEFiLM Watch Movies Online	Amazon Wireless Cellphones & Wireless Plans	Junglee India Online Shopping	DPReview Digital Photography	Audible Download Audio Books

The table uses compact URIs (CURIEs) to refer to source vocabularies. The CURIE *schema* refers to the Schema.org vocabulary. The CURIE *mo* refers to the Movie Ontology vocabulary. The CURIE *owl* refers to the Web Ontology Language (OWL) specification. The CURIE *rdf* refers to the RDF specification. The CURIE *rdfa* refers to the RDFa specification.

Table 9: Mapping Correspondences for IMDb Using the Baseline Approach

External model term	Internal model term	Confidence measure
schema:Movie	mo:Movie	1
schema:director	mo:hasDirector	0.842105263
schema:actor	mo:hasActor	0.769230769
schema:genre	mo:isGenreOf	0.714285714
schema:actors	mo:hasActor	0.714285714
schema:awards	mo:isAwardOf	0.666666667
schema:bestRating	mo:imdbrating	0.6
schema:ratingValue	mo:imdbrating	0.571428571
schema:ratingCount	mo:imdbrating	0.571428571
schema:contentRating	mo:imdbrating	0.52173913
schema:aggregateRating	mo:imdbrating	0.48
schema:name	mo:isGenreOf	0.461538462
schema:name	mo:isGenreOf	0.461538462
schema:Person	mo:Sound_Mix	0.461538462
schema:duration	mo:imdbrating	0.444444444
md:item	mo:title	0.444444444
schema:image	owl:members	0.428571429
#mentions	mo:runtime	0.4
#comment	rdf:rest	0.4
schema:trailer	rdf:rest	0.4
schema:reviewCount	mo:containsCountry	0.384615385
schema:audience	mo:isConsumableAs	0.384615385
schema:provider	mo:produced	0.375
schema:writer	mo:title	0.363636364
schema:datePublished	mo:releasedate	0.333333333
schema:description	mo:enablesConsumptionOf	0.322580645
schema:MusicRecording	owl:Thing	0.315789474
schema:AggregateRating	owl:Thing	0.3
schema:url	rdf:rest	0.285714286
schema:keywords	mo:hasColor	0.25
schema:headline	mo:isActorIn	0.235294118
rdfa:usesVocabulary	rdf:rest	0.222222222
#musicBy	mo:isActressIn	0.222222222

8.5.2 IMDb – The Experimental Approach

With the experimental approach, five sets of mapping correspondences were created: one set for each strategy and one set for all of the strategies combined. To provide an overview of the results, 15 example mapping correspondences from the combined strategy set are shown in Table 10. The full set of results can be found in the DVD accompanying this thesis.

Table 10: Example Mapping Correspondences from Combined Strategy Results

External model term	Internal model term	Confidence measure
strat2:runtime	mo:runtime	1
strat2:director	mo:hasDirector	0.842105263
strat2:color	mo:hasColor	0.769230769
schema:genre	mo:isGenreOf	0.714285714
strat4:titlecast	mo:title	0.714285714
strat2:genres	mo:isGenreOf	0.666666667
schema:awards	mo:isAwardOf	0.666666667
strat2:country	mo:containsCountry	0.636363636
schema:Movie	mo:Film-Noir	0.615384615
strat2:release_date	mo:releasedate	0.608695652
strat2:production_co	mo:produced	0.571428571
strat1:title_trailer	mo:title	0.555555556
strat4:ratingwidget	mo:imdbrating	0.545454545
strat4:titlestoryline	mo:title	0.526315789
strat4:title_overview	mo:title	0.526315789
…	…	…

It can be seen that the added properties provide matches with high confidence measure. However, it was found that there are properties that took over the original Schema.org terms and resulted in less relevant matches as well.

The result mapping correspondences for the rest of the dataset can be found on the DVD accompanying this thesis.

8.6 Analysis

This section presents an analysis on the experimental results. The section discusses the chosen metrics, described in sub-section 8.3.4, for each dataset. The section concludes with an overall analysis of the experiment.

The statistical analysis is shown below in detail for the IMDb dataset. The statistical analysis for the rest of the dataset is summarised in Table 11.

IMDb – Baseline

The total number of HTML elements of this web page is 1,612 and the total number of terms in the internal model is 120. This makes the total number of every possible matching outcome to be 193,440.

Comparing to the gold standard, the baseline approach results in the following:

- True positive: 5 matches
- False positive: 33 matches
- True negative: 193,391 matches, and
- False negative: 11 matches.

This results in a precision of 13.16%, a recall of 31.25% and an F-measure score of 0.18.

IMDb – Experimental – Strategy#1

Comparing to the gold standard, the experimental approach using Strategy#1 results in the following:

- True positive: 2 matches
- False positive: 31 matches
- True negative: 193,395 matches, and
- False negative: 14 matches.

This results in a precision of 6.06%, a recall of 12.05% and an F-measure score of 0.08.

IMDb – Experimental – Strategy#2

Comparing to the gold standard, the experimental approach using Strategy#2 results in the following:

- True positive: 7 matches
- False positive: 54 matches
- True negative: 193,370 matches, and
- False negative: 9 matches.

This results in a precision of 11.48%, a recall of 43.75% and an F-measure score of 0.18.

IMDb – Experimental – Strategy#3

Comparing to the gold standard, the experimental approach using Strategy#3 results in the following:

- True positive: 1 match
- False positive: 15 matches
- True negative: 193,409 matches, and
- False negative: 15 matches.

This results in a precision of 6.25%, a recall of 6.25% and an F-measure score of 0.06.

IMDb – Experimental – Strategy#4

Comparing to the gold standard, the experimental approach using Strategy#4 results in the following:

- True positive: 1 match
- False positive: 31 matches
- True negative: 193,395 matches, and
- False negative: 14 matches.

This results in a precision of 6.06%, a recall of 12.05% and an F-measure score of 0.08.

IMDb – Experimental – Combined strategy

Comparing to the gold standard, the experimental approach using the combination of all strategies results in the following:

- True positive: 7 matches
- False positive: 106 matches
- True negative: 193,318 matches, and
- False negative: 9 matches.

This results in a precision of 6.19%, a recall of 43.75% and an F-measure score of 0.11.

For the rest of the dataset, Table 11 summarises the statistical metrics.

Mean and median metric values achieved by individual approaches are shown in Table 12 and Table 13 respectively.

It is evident from the experimental results that the extraction strategies in the external model enrichment process do not necessarily improve precision and recall. Moreover, combining multiple strategies does not result in a better performance.

However, it might be worth noting that among the true positive matches of both approaches, properties created from Strategy#2 (heading elements) often have higher confidence values. It is also worth noting that; due to the simplicity of the matching algorithm, “directions” of properties are lost. For example, the algorithm found *schema:genre*, a property for relating a film to a genre, to be a good match for *mo:isGenreOf*, which relates a genre to a film. It might be expected that a more sophisticated matching algorithm may yield better results.

By observing the results, areas of improvements were found in software implementation. It was found that additional terms from the external model enrichment process usually take over the existing semantic markup terms, reducing or at times discarding correct matches. It was also found that the software implementation was unable to retain semantic markup terms that were nested in a hierarchy of terms. Therefore, correct matches were discarded.

Table 11: Summary of Statistical Metrics for Each Dataset

Dataset	Approach	Precision (%)	Recall (%)	F-measure
Rotten Tomatoes	Baseline	20.00	62.50	0.30
	Strategy#1	6.67	12.50	0.09
	Strategy#2	1.89	12.50	0.03
	Strategy#3	9.09	12.50	0.10
	Strategy#4	6.45	25.00	0.10
	Combined	2.50	25.00	0.04
Barnes & Noble	Baseline	4.54	11.11	0.06
	Strategy#1	2.94	11.11	0.05
	Strategy#2	3.51	22.22	0.06
	Strategy#3	14.28	11.11	0.12
	Strategy#4	2.82	22.22	0.05
	Combined	1.34	11.11	0.02
Goodreads	Baseline	4.54	14.28	0.07
	Strategy#1	2.32	14.28	0.04
	Strategy#2	6.67	14.28	0.09
	Strategy#3	9.09	14.28	0.11
	Strategy#4	2.78	14.28	0.05
	Combined	1.33	14.28	0.02
MySpace	Baseline	33.33	40.00	0.36
	Strategy#1	50.00	10.00	0.17
	Strategy#2	14.28	10.00	0.12
	Strategy#3	66.67	20.00	0.31
	Strategy#4	25.00	10.00	0.14
	Combined	20.00	20.00	0.20
Last.fm	Baseline	36.84	70.00	0.48
	Strategy#1	22.22	20.00	0.21
	Strategy#2	16.13	50.00	0.24
	Strategy#3	42.86	30.00	0.35
	Strategy#4	7.50	30.00	0.12
	Combined	7.14	50.00	0.12

Table 12: Mean Metric Values of Individual Approaches

Approach	Mean precision (%)	Mean recall (%)	Mean F-measure
Baseline	18.74	38.19	0.24
Strategy#1	15.04	13.40	0.10
Strategy#2	8.99	25.46	0.12
Strategy#3	24.71	15.69	0.18
Strategy#4	7.75	17.96	0.08
Combined	6.42	29.21	0.09

Table 13: Median Metric Values of Individual Approaches

Approach	Median precision (%)	Median recall (%)	Median F-measure
Baseline	16.58	35.62	0.24
Strategy#1	6.36	12.50	0.08
Strategy#2	9.07	18.25	0.10
Strategy#3	11.67	13.39	0.12
Strategy#4	4.63	18.25	0.08
Combined	4.35	23.61	0.08

8.7 Conclusion

The analysis of the experiment results has not provided evidence that the external model enrichment process improves precision and recall of web data extraction. However, possible improvements on the external model enrichment process have been identified as follows:

Prioritisation in external model enrichment

It was evident that the added properties may replace existing properties that are already suitable matches. This can be improved by prioritising the properties in merging the original external model and the added properties. It is expected that this improvement would increase precision and recall.

More extraction strategies

By the design of the external model enrichment process, there are possibilities of adding more extraction strategies. It is expected that the added extraction strategies would increase the amount of data in the enriched external model and therefore increase recall.

Such an extraction strategy is detection of key-value data patterns. HTML elements that represent key-value data are found by the author to have similar CSS. These HTML elements are often aligned in a tabular manner, regardless of the use of the HTML table element, with distinguishable rows and columns. A hypothesis can be made that key-value data on a single-instance web page describes fragments of the instance.

More sophisticated matching algorithms

The external model enrichment process was designed to be independent from the matching algorithm. Therefore, a different and possibly more sophisticated matching algorithm could be “plugged in”, as long as it supported the Alignment format. Such algorithms may take into account other information, in addition to similarity of URIs, such as data types and hierarchy of terms.

The next chapter presents the overall conclusion of this thesis.

9 CONCLUSION

This chapter begins with a section reflecting on the research objectives and their corresponding achievements. The following section outlines an assessment of the contributions this thesis has made to web extraction and semantic markup research. Next, section 9.3 discusses possibilities for future work based on the research carried out to date. Finally, section 9.4 concludes the thesis with final remarks.

9.1 RESEARCH OBJECTIVES

The following discussion is corresponding to five research objectives described in section 1.3.

RO1: To conduct a survey on state-of-the-art research on web data extraction, applications of semantic markup, and semantic mapping.

The survey found that the state-of-the-art web data extraction techniques are based on the concept of *wrappers*. Wrappers are programs that process data from web sources in such a way that database-like queries can be performed on the data. Current wrappers use HTML source code and document object model (DOM) representations of the source code as inputs. These inputs are processed using algorithms such as pattern recognition or algorithms based on tree data structures.

Although semantic markup is found to be gaining popularity as an approach to publishing data on the web, there is little acknowledgement of semantic markup in web data extraction. The survey also found current challenges in web data extraction. The survey concluded that using semantic markup in web data extraction may help address two challenges: reducing human efforts and uniform data structure.

RO2: To design and implement a semantic markup mapping tool that exploits the semantic markup's links to HTML elements.

A semantic markup mapping tool was designed and implemented. The tool's design was based on a set of use cases derived from the state-of-the-art cognitive support framework for manual semantic mapping. The tool provides visual clues to highlight links between semantic markup and annotated HTML elements.

The semantic markup mapping tool was implemented as a Google Chrome web browser extension. The tool uses Alignment API to generate mapping correspondence candidates.

RO3: To conduct an evaluation on the semantic markup mapping tool.

A user experiment was conducted to evaluate the semantic markup mapping tool. Participants were categorised by their expertise into three groups: domain experts, IT personnel and end users. The experiment was aimed to find if the tool helps reduce gap in expertise, resulting in similar performance: speed and accuracy. The experiment also included a questionnaire to measure perceived usefulness of the links between semantic markup data and HTML elements, and usability of the tool.

The experiment results provided evidence that expertise in semantic web technology still correlates with better speed and accuracy. The results did not contain evidence that expertise in information technology contribute to speed or accuracy in using the tool.

It was concluded from the results showed that more training in using the tool could help improve accuracy. It was also evident in the questionnaire feedback that the links between semantic markup data and HTML elements were perceived as useful.

RO4: To design and implement a process for extracting web data using semantic markup and the DOM.

Following the results from the evaluation of the semantic markup mapping tool, the external model enrichment process was developed. The external model enrichment process is a process of adding additional properties from data in the document object model representation of a web page to the external model created from semantic markup data. The external model enrichment process was developed based on the assumption that the added properties could improve precision and recall of web data extraction.

The enrichment process uses four extraction strategies to add additional properties. The first strategy, Strategy#1, creates properties from values in attributes *class* and *id* in HTML elements. The second strategy, Strategy#2 creates properties from values in heading HTML elements. The third strategy, Strategy#3, creates properties from presence of embedded content such as images and videos. The fourth strategy, Strategy#4, creates properties from values in attributes class and id in the hierarchy between annotated HTML elements.

The external model enrichment process was implemented as a Google Chrome web browser extension. The web browser extension uses the set of extraction strategies to create an enriched external model. The enriched external model can be serialised into a file for mapping.

RO5: To conduct an evaluation on the process for extracting web data using semantic markup and the DOM.

An experiment was conducted to evaluate the performance of the external model enrichment process. The metrics used in the evaluation were precision, recall and the F-measure score. Experiment results were obtained using two approaches: the baseline approach and the experimental approach. With the baseline approach, external models are created from only semantic markup data. With the experimental approach, external models are created from semantic markup data and addition properties from the external model enrichment process. External models from both approaches are mapped to corresponding internal models using the Alignment API. Mapping correspondence sets from both approaches are compared to the gold standard mapping correspondences. The gold standard was created manually to represent the ideal web data extraction results created only by human.

The results showed that the f-measure score increased by 16.67%. This was caused by a 6.62% decrease in precision and 4.25% increase in recall. The enriched external model contains more properties with high mapping confident values. This is probably because there are more choices of property terms for the Alignment API to choose from.

Possible improvements for the external model enrichment process have been identified as **prioritisation in external model enrichment**, **more extraction strategies** and **more sophisticated matching algorithms**.

9.2 CONTRIBUTIONS

This chapter presents a discussion on contributions of this thesis.

9.2.1 MAJOR CONTRIBUTION – THE EXTERNAL MODEL ENRICHMENT PROCESS

A major contribution of this thesis is the proposed ***external model enrichment process***. The external model enrichment process was designed to improve web data extraction in terms of precision and recall by adding properties to the external model. This was motivated by the growing popularity of semantic markup, whereas a survey on state-of-the-art research shows that there has been little research on using semantic markup in web data extraction.

The external model enrichment process has been evaluated in an experiment described in chapter 8. Possible improvements to the external model enrichment process have also been identified.

9.2.2 MINOR CONTRIBUTION – THE SEMANTIC MARKUP MAPPING TOOL

A minor contribution of this thesis is the design of the semantic markup mapping tool. The semantic markup mapping tool was designed and implemented to explore the possibilities of using semantic markup in web data extraction. The tool assists a manual mapping process by highlighting HTML elements annotated with semantic markup data. A user experiment on the tool was conducted with participants with different levels of expertise in semantic web technologies. The tool proved to be useful to participants who are most familiar with semantic web technologies. The results from the user experiment also showed that a major part of the mistakes users make are trivial and could be eliminated with more training.

The early design of the semantic markup mapping tool has been published in the following *breaking news* abstract paper:

> Seeoun, T., Brennan, R., & O'Sullivan, D. (2011). User-Centric Mapping for RDFa Web Mining. The 1st International Conference on Web Intelligence, Mining and Semantics (WIMS'11).

9.2.3 MINOR CONTRIBUTION – SOFTWARE PROTOTYPES

Another minor contribution of this thesis is the software prototypes that have been implemented. Two Google Chrome web browser extensions have been created for evaluations. One was created to evaluate the design of the semantic markup mapping tool.

Another was created to evaluate the external model enrichment process. Both tools were developed in widely used web technology: HTML5, JavaScript, and CSS. In the author’s opinion, this makes the software prototypes easy to be learnt from and developed further.

9.3 Future Work

With the external model enrichment process, it is possible to use the enriched external model in manual mapping tasks. This prompts an investigation on whether the enriched external model can be used to create a more relevant set of candidate mapping correspondences for the semantic markup mapping tool.

Moreover, it has been pointed out in section 8.7 the areas where the external model enrichment process can be improved. The suggested improvements can be implemented and evaluated based on the current dataset.

Evaluation of the external model enrichment process on a broader dataset would also enable a more generalised analysis.

9.4 Final Remarks

The author believes that high level of accuracy in web data extraction will always require reviewing by human. Therefore, it is important to not exclude human from a design of a web data extraction tool.

Data publishing technologies evolve in a fast pace. For example, Microdata’s rate of adoption, in terms of websites, seems to have increased to almost match RDFa within a year. Therefore, research in web data extraction should be fast to investigate new sources of data.

The author also believes that new publishing technologies that blur the line between machine-readable and human-readable data will benefit web data extraction greatly.

BIBLIOGRAPHY

Abiteboul 1997 Abiteboul, S. (1997). Querying semi-structured data. *Database Theory—ICDT'97*, 1-18.

Adida et al. 2008 Adida, B., Birbeck, M., McCarron, S., & Pemberton, S. (2008). RDFa in XHTML: Syntax and processing. *Recommendation, W3C*.

Al-Jabari et al. 2009 Al-Jabari, M., Mrissa, M., & Thiran, P. (2009). Towards web usability: Providing web contents according to the readers contexts. *User Modeling, Adaptation, and Personalization*, 467-473.

Aumueller et al 2005 Aumueller, D., Do, H. H., Massmann, S., & Rahm, E. (2005). Schema and ontology matching with COMA++. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 906-908.

Baumgartner et al. 2005 Baumgartner, R., Frölich, O., Gottlob, G., Harz, P., Herzog, M., Lehmann, P., & Wien, T. U. (2005). Web data extraction for business intelligence: the Lixto approach. *Datenbanksysteme in Business, Technologie und Web*, 11, 30-47.

Baumgartner et al. 2009 Baumgartner, R., Gatterbauer, W., & Gottlob, G. (2009). Web data extraction system. *Encyclopedia of Database Systems*, 3465-3471.

Berners-Lee 2006 Berners-Lee, T. (2006). Linked data-design issues.

Berners-Lee et al. 2009 Berners-Lee, T., Fielding, R., & Masinter, L. (1998). RFC 2396: Uniform resource identifiers (URI): Generic syntax.

Bossa et al. 2006 Bossa, S., Fiumara, G., & Provetti, A. (2006). A lightweight architecture for rss polling of arbitrary web sources. In *Proceedings of workshop from objects to agents (WOA)*.

Bizer et al. 2009	Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3), 1-22.
Chang et al. 2006	Chang, C., Kayed, M., Girgis, M. R., & Shaalan, K. F. (2006). A survey of web information extraction systems. *IEEE transactions on knowledge and data engineering*, 18(10), 1411.
Cruz et al. 2009	Cruz, I. F., Antonelli, F. P., & Stroe, C. (2009). AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment*, 2(2), 1586-1589.
Coombs et al. 1987	Coombs, J. H., Renear, A. H., & DeRose, S. J. (1987). Markup systems and the future of scholarly text processing. *Communications of the ACM*, 30(11), 933-947.
Dadzie and Rowe 2011	Dadzie, A. S., & Rowe, M. (2011). Approaches to visualising Linked Data: A survey. *Semantic Web*, 2(2), 89-124.
Eikvil 1999	Eikvil, L. (1999). Information extraction from world wide web- a survey. *Norwegian Computing Center*, (945).
Embley et al. 1999	Embley, D. W., Campbell, D. M., Jiang, Y. S., Liddle, S. W., Lonsdale, D. W., Ng, Y. K., & Smith, R. D. (1999). Conceptual-model-based data extraction from multiple-record Web pages. *Data & Knowledge Engineering*, 31(3), 227-251.
Euzenat 2004	Euzenat, J. (2004). An API for ontology alignment. *The Semantic Web – ISWC 2004*, 698-712.
Euzenat and Shvaiko 2007	Euzenat, J., & Shvaiko, P. (2007). *Ontology matching* (Vol. 18). Berlin: Springer.
Falconer et al. 2006	Falconer, S. M., Noy, N. F., & Storey, M. A. (2006). Towards understanding the needs of cognitive support for ontology mapping. *Ontology Matching*, 225.

Falconer and Storey 2007	Falconer, S., & Storey, M. A. (2007). A cognitive support framework for ontology mapping. *The Semantic Web*, 114-127.
Ferrara et al. 2012	Ferrara, E., De Meo, P., Fiumara, G., & Baumgartner, R. (2012). Web Data Extraction, Applications and Techniques: A Survey. arXiv preprint arXiv:1207.0246.
Fiumara 2007	Fiumara, G. (2007). Automated information extraction from web sources: a survey. Between Ontologies and Folksonomies, 1.
Fu 2011	Fu, B. (2011). *Semantic-Oriented Cross-Lingual Ontology Mapping* (Doctoral thesis), Trinity College Dublin, Dublin, Ireland.
Furche et al. 2012	Furche, T., Gottlob, G., Grasso, G., Gunes, O., Guo, X., Kravchenko, A., ... & Wang, C. (2012, April). DIADEM: domain-centric, intelligent, automated data extraction methodology. In *Proceedings of the 21st international conference companion on World Wide Web*, ACM, 267-270.
Geel et al. 2012	Geel, M., Church, T., & Norrie, M. C. (2012, September). Sift: an end-user tool for gathering web content on the go. *In Proceedings of the 2012 ACM symposium on Document engineering* (pp. 181-190). ACM.
Goyal and Westenthaler 2004	Goyal, S., & Westenthaler, R. (2004). RDF Gravity (RDF Graph Visualization Tool). Salzburg Research, Austria.
Granitzer et al. 2010	Granitzer, M., Sabol, V., Onn, K. W., Lukose, D., & Tochtermann, K. (2010). Ontology alignment – a survey with focus on visually supported semi-automatic techniques. *Future Internet*, 2(3), 238-258.
Heflin et al. 1999	Heflin, J., Hendler, J., & Luke, S. (1999). SHOE: A knowledge representation language for internet applications, *Technical*

Report, CS-TR-4078 (UMIACS TR-99-71), Dept. of Computer Science, University of Maryland.

Hickson 2011 Hickson, I. (2011). HTML microdata, *Working Draft*, W3C.

Hu et al. 2008 Hu, W., & Qu, Y. (2008). Falcon-AO: A practical ontology matching system. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3), 237-239.

Kalfoglou and Schorlemmer 2003 Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The knowledge engineering review*, 18(1), 1-31.

Khalili and Auer 2012 Khalili, A., & Auer, S. (2012). User interfaces for semantic content authoring: a systematic literature review. Leipzig, Germany: Agile Knowledge Engineering and Semantic Web (AKSW) research group. Retrieved October 2012, from http://svn.aksw.org/papers/2011/JWS_SemanticContentAuthoring/public.pdf

Khalili et al. 2012 Khalili, A., Auer, S., & Hladky, D. (2012). The rdfa content editor-from WYSIWYG to WYSIWYM. *In IEEE Signature Conference on Computers, Software, and Applications*, COMPSAC (Vol. 2012).

Khare and Çelik 2006 Khare, R., & Çelik, T. (2006). Microformats: a pragmatic path to the semantic web. *In Proceedings of the 15th international conference on World Wide Web* (pp. 865-866). ACM.

Laender et al. 2002 Laender, A. H., Ribeiro-Neto, B. A., da Silva, A. S., & Teixeira, J. S. (2002). A brief survey of web data extraction tools. *ACM Sigmod Record, 31*(2), 84-93.

Lassila and Swick 1998 Lassila, O., & Swick, R. R. (1998). Resource description framework (RDF) model and syntax specification.

Luczak-Rösch and Heese 2009	Luczak-Rösch, M., & Heese, R. (2009). Linked data authoring for non-experts. *In Workshop on Linked Data on the Web*.
Makhoul et al. 1999	Makhoul, J., Kubala, F., Schwartz, R., & Weischedel, R. (1999). Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop* (pp. 249-252).
Mika and Potter 2012	Mika, P., & Potter, T. (2012). Metadata statistics for a large web corpus. In *Proceedings of the Linked Data Workshop (LDOW) at the International World Wide Web Conference*.
Mühleisen and Bizer 2012	Mühleisen, H., & Bizer, C. (2012). Web Data Commons Extracting Structured Data from Two Large Web Corpora. *Invited paper at the 5th Linked Data on the Web*.
Noy et al. 2001	Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R. W., & Musen, M. A. (2001). Creating semantic web contents with protege-2000. *Intelligent Systems,* IEEE, 16(2), 60-71.
Plumbaum et al. 2012	Plumbaum, T., Lommatzsch, A., De Luca, E., & Albayrak, S. (2012). Serum: Collecting semantic user behavior for improved news recommendations. *Advances in User Modeling*, 402-405.
Segaran et al. 2009	Segaran, T., Evans, C., & Taylor, J. (2009). *Programming the semantic web*. O'Reilly Media, Incorporated.
Shvaiko and Euzenat 2013	Shvaiko, P.; Euzenat, J. (2013). Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering, 25(1), 158-176.
Steiner et al. 2010	Steiner, T., Troncy, R., & Hausenblas, M. (2010). How Google is using linked data today and vision for tomorrow. *Linked Data in the Future Internet*.

Tramp et al. 2010 Tramp, S., Heino, N., Auer, S., & Frischmuth, P. (2010). RDFauthor: employing RDFa for collaborative knowledge engineering. *Knowledge Engineering and Management by the Masses*, 90-104.

Willighagen and Wikberg 2010 Willighagen, E. L., & Wikberg, J. E. (2010). Linking open drug data to cheminformatics and proteochemometrics. *In SWAT4LS-2009-Semantic Web Applications and Tools for Life Sciences*, 559.

Wong and Hong 2007 Wong, J., & Hong, J. I. (2007). Making mashups with marmite: towards end-user programming for the web. *In Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1435-1444). ACM.

APPENDIX

This appendix presents additional details for this thesis. The appendix contains the following information:

- Details of the content on the accompanying DVD.
- Instruction sheets, titled “Evaluation of a User-Centric RDFa Mapping Tool”, for the semantic markup mapping tool user experiment discussed in chapter 6.
- Questionnaire, titled “Post Experiment Questionnaire for Evaluation of a User-Centric RDFa Mapping Tool”, used in the semantic markup mapping tool user experiment.

Please note that terminology in these documents may differ slightly from that in this thesis.

DVD Content

The DVD accompanying this thesis contains source code for software implementation and experiment data. The DVD content is organised as follows:

Folder name	Content
Semantic markup mapping tool	
• Implementation	Source code of a Google Chrome web browser extension.
• Evaluation	Raw data from the user experiment.
External model enrichment process	
• Implementation	Source code of a Google Chrome web browser extension which contains extraction strategies.
• Evaluation	CSV and RDF files of external and internal models used in the experiment.

Evaluation of a User-Centric RDFa Mapping Tool

Background

Welcome to the usability evaluation of a user-centric RDFa mapping tool. If you are familiar with semantic web technologies or at least with RDF, you may skip this section. Otherwise, please read the following brief explanation. You may ask the experimenter at any time.

RDF stands for Resource Description Framework. It is a way to describe information in a form of “triples” or groups of three terms, a subject, a property and an object, tied together. Each term represents a concept. Let’s take a look at triples describing the Guinness stout.

Subject	Property	Object
Guinness	is_a	Stout
Stout	is_a	Beer
Stout	has_colour	Black

We may conclude from these triples that Guinness is a stout, which is a kind of dark beer. This, however, is possible if we define some “rules” for specific terms because ultimately each term is just a set of characters.

For example, the property “has_colour” would be quite meaningless if we did not define that the objects associated to it must be of type “Colour”.

Now, what is RDFa? The “a” here stands for “attribute” which means attributes in XHTML tags. If you are not familiar with XHTML, it is a language for creating web pages. Certain terms are predefined so web browsers can show a web page the way the designer intended. For example, writing “<br/>” would result in a line break.

The “X” in XHTML stands for “extensible”. This means we can add custom attributes into our web pages. Using RDFa means we use some existing XHTML attributes and some new attributes to add RDF data into our web pages.

For example, if we want to write our name on a web page we created, we may write

<span>John Doe</span>

but the computers reading this web page code would not be able to know that it is an author’s name (actually they do not even know that it is a name). Using RDFa, we can add an attribute “property” to tell the computer that that particular name is the author’s name by writing

<span property=”author”>John Doe</span>

Similar to the example Guinness triples above, the property “author” here is meaningless unless people agree upon its definition. There have been many initiatives that define sets of “schemas” for other people to reuse. These vocabulary sets are represented not by words but by identifiers or pointers to their definitions called “URI” (of which the details will not be discussed here). These addresses are often abbreviated into “prefixes”. From the example above, instead of using our very own property “author” we may write

<span property=”dc:creator”>John Doe</span>

which uses the term “creator” from a vocabulary set called “Dublin Core” (hence the prefix “dc”). This ensures that most of the people who use this vocabulary set will understand the role John Doe is playing here: he is the creator of the web page.

Schema Mapping

Schema mapping is a process of mapping terms between two schemas that describe things. Let's take a look at two class schemas below:

Thing	Thing
- Department	- Faculty
- Person	- Member
- - Lecturer	- - Professor
- - Student	- - Student
- - - UndergraduateStudent	- - - BAStudent
- - - PostgraduateStudent	- - - MAStudent
- - - - MastersStudent	- - - PhDStudent
- - - - DoctoralStudent	

You can see that despite the difference in structure, we can easily compare and map terms from one schema to another. Mapping them will help in merging data from both schemas.

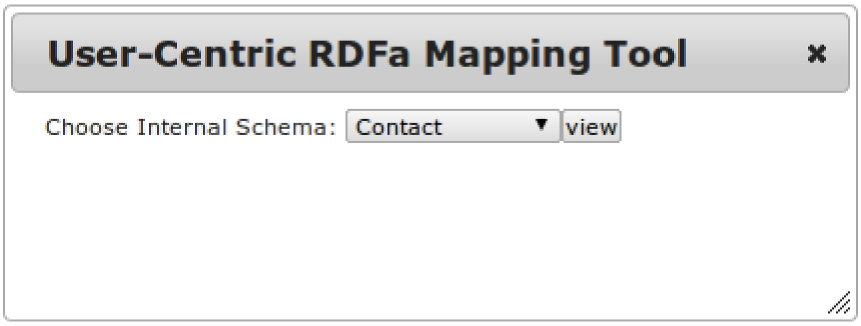
In this work, we will map a schema obtained from a web page that has RDFa data to a predefined schema of our own which will be called an “internal schema”.

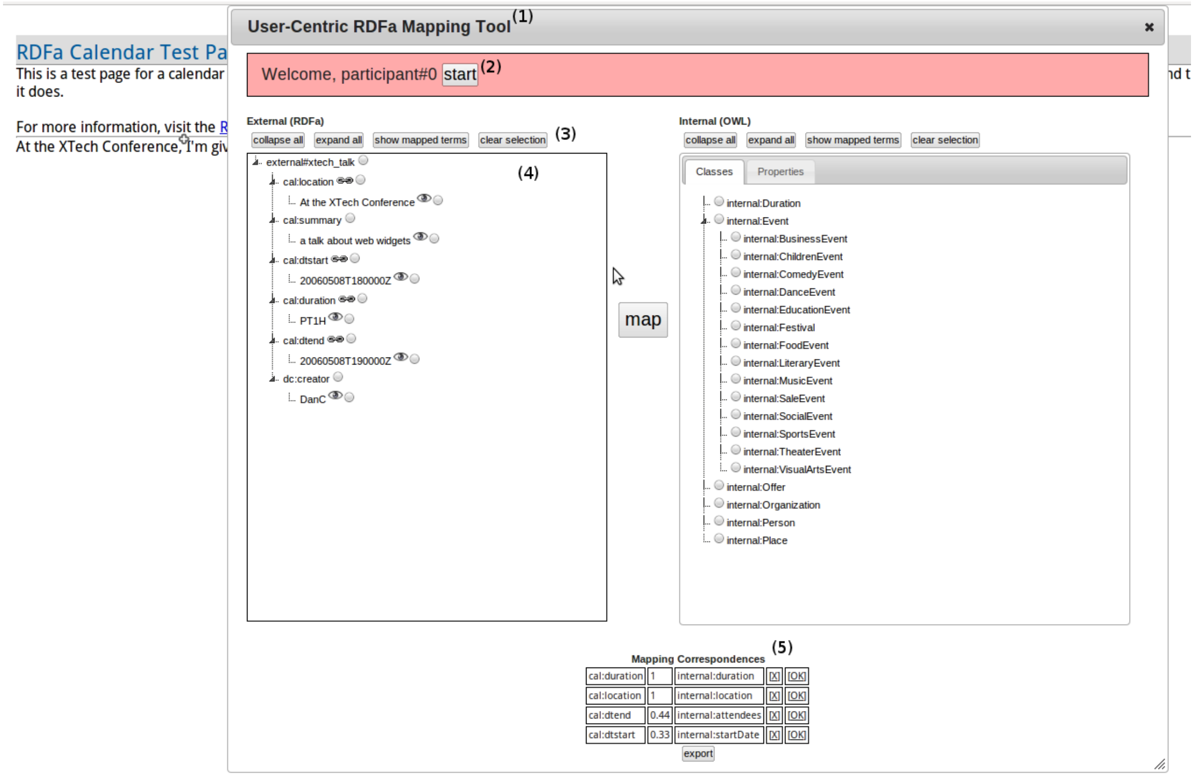
RDFa Mapping Tool

Now let’s take a look at what you will be using. The tool prototype is implemented as a Google Chrome web browser extension. If it detects RDFa data in the web page, it will show a box icon like this:

in the bottom right corner of the page.

Let’s try this functionality by clicking on the first bookmark. It will take you to a simple event calendar page. Click on the box icon to toggle the tool's interface. You will first be asked for the target internal schema, so here we choose “Event” and click on “view”.





Mapping Correspondences

cal:duration	1	internal:duration	[X]	[OK]
cal:location	1	internal:location	[X]	[OK]
cal:dtend	0.44	internal:attendees	[X]	[OK]
cal:dtstart	0.33	internal:startDate	[X]	[OK]

The main interface should now appear. Let's take a look at each component.

1. **Dialogue** – You can move the dialogue around by dragging the title bar. You can also resize it at borders and corners. The dialogues can be closed by clicking the X mark on the top right corner (or pressing Esc) and reopened by clicking the icon mentioned above.

2. **Task** – Here the tasks that you will have to do are displayed. We will come back to this part later on.

3. **Schema Display Control** – Schemas from both sources are displayed as collapsible trees. For each tree, there are four buttons that control the whole schema: "collapse all", "expand all", "show mapped terms" and "clear selection".

4. **Schema Tree** – Both trees consist of “terms” taken from various vocabularies which represent concepts that are being used.

   4.1. External Schema on the left hand side shows triples extracted from the current web page. Its hierarchy distinguishes roles of each term, the first level being subject, the second level being properties, the third level being objects, the fourth level being properties, and so on. The following is an example of an External Schema.

   ```
   http://www.example.com#me (subject)
   - foaf:name (property)
   - - “Tewson Seeoun” (object)
   ```

   4.2. Internal Schema on the right hand side shows classes and properties in the target schema of mapping. Hierarchy of the class tab represents subclass relationships. There is not any hierarchy on the property tab. Hovering the cursor over each property displays its URI, domain and range.

   Clicking on a small triangle in front of terms collapses or expands their child nodes. Each term can be selected by clicking on the term itself or on the radio button displayed after it.

   Clicking on the chain icon (🔗) shows terms in another schema that are mapped (either automatically or manually) to them.

   If an eye icon (👁) is displayed after a term, it means that triple is associated with a visible HTML element. Clicking on the icon locates that element on the page.

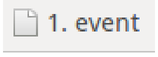
   Because the tool overlays the web page, some interaction you would do to a web page can be done to the tool, too. For example, you can press Ctrl+F to search for terms in the schemas or click and drag the cursor to select text.

5. **Mapping Correspondences** – or mapping pairs. You can remove a mapping pair by clicking on “X” or approve the pair by clicking on “OK”. Numbers in the second column are confidence values generated from automatic mapping. The higher the values are, the higher confidence the mapping pairs have. Approval of a pair changes the value to 1.

Please try to play around and get familiar with the user interface. Then we will proceed to the experiment.

Tasks

1. Click on the first bookmark to open an RDFa Calendar Test web page. This page contains RDFa data about an event, e.g., location and time. Open the mapping user interface and choose “Event” as target schema. Click on “Start” and finish four given tasks shown in the dialogue.



1. event

2. Click on the second bookmark to open a web page describing personal contact information of Mark Birbeck. Open the mapping user interface. Choose “Contact” as the target schema and click on “Start” for instructions. The gold standard table below will be used in the task.

2. contact

External	Internal
foaf:Person	internal:PersonContact
foaf:name	internal:nickname
foaf:weblog	internal:url
foaf:holdsAccount	internal:url
foaf:OnlineAccount	Internal:IMAccount

3. Click on the third bookmark to open an online shopping web page and repeat this task following the table below

External	Internal
v:name	internal:name
vcard:Address	internal:address
vcard:latitude	internal:geo
vcard:longtitude	internal:geo
foaf:account	internal:url
gr:hasOpeningHoursDayOfWeek	internal:openingHours

4. Click on the fourth bookmark to open a BBC Music web page. Open the tool's interface and select an appropriate target internal schema. Click “Start” for instructions. Repeat the task on the fifth bookmark which opens a product web page of an iPad2 case.

Post Experiment Questionnaire for Evaluation of a User-Centric RDFa Mapping Tool

How would you rate your familiarity in using the internet through web browsers?

	1	2	3	4	5	
Unfamiliar at all						Most familiar

How would you rate your knowledge in semantic web technology in general?

	1	2	3	4	5	
None						Very competent

How would you rate the overall usability of the prototype?

	1	2	3	4	5	
Obstructively complicated						Very useful

How would you rate the appropriateness of the tree display?

	1	2	3	4	5	
Most inappropriate						Most appropriate

How would you rate the appropriateness of the collapsible tree navigation?

	1	2	3	4	5	
Most inappropriate						Most appropriate

What is the most useful feature of the prototype?

What is the most confusing feature of the prototype?

Please suggest features that would improve the prototype.

End of questionnaire.