Automatic Extraction of Semantically-Meaningful Information from the Web.

J.L. Arjona, R. Corchuelo, A. Ruiz, and M. Toro

Escuela Técnica Superior de Ingenieŕia Informática de la Universidad de Sevilla
Departamento de Lenguajes y Sistemas Informáticos
Avda. de la Reina Mercedes, s/n, Sevilla (SPAIN)
{arjona,corchu,aruiz,mtoro}@lsi.us.es

Abstract. The semantic Web will bring meaning to the Internet, making it possible for web agents to understand the information it contains. However, current trends seem to suggest that the semantic web is not likely to be adopted in the forthcoming years. In this sense, meaningful information extraction from the web becomes a handicap for web agents. In this article, we present a framework for automatic extraction of semantically-meaningful information from the current web. Separating the extraction process from the business logic of an agent enhances modularity, adaptability, and maintainability. Our approach is novel in that it combines different technologies to extract information, surf the web and automatically adapt to web changes.

Keywords: *Web agents, information extraction, wrappers, and ontologies.*

1 Introduction

In recent years, the web has consolidated as one of the most important knowledge repositories. Furthermore, the technology has evolved to a point in which sophisticated new generation web agents proliferate. They enable e cient, precise, and comprehensive retrieval and extraction of information from the vast web information repository. They can also circumvent some problems related to slow Internet access, and free up prohibitively expensive surf time by operating in the background. It is thus not surprising that web agents are becoming so popular.

A major challenge for web agents has become sifting through an unwieldy amount of data to extract meaningful information. Two important factors contribute to these di culties: rst, the information on the web is mostly available in human-readable forms that lack formalised semantics that would help agents

[*] The work reported in this article was supported by the Spanish Inter-ministerial Commission on Science and Technology under grant TIC2000-1106-C02-01

use it [3]; second, the information sources are likely to change their structure, which usually has an impact on their presentation but not on their semantics.

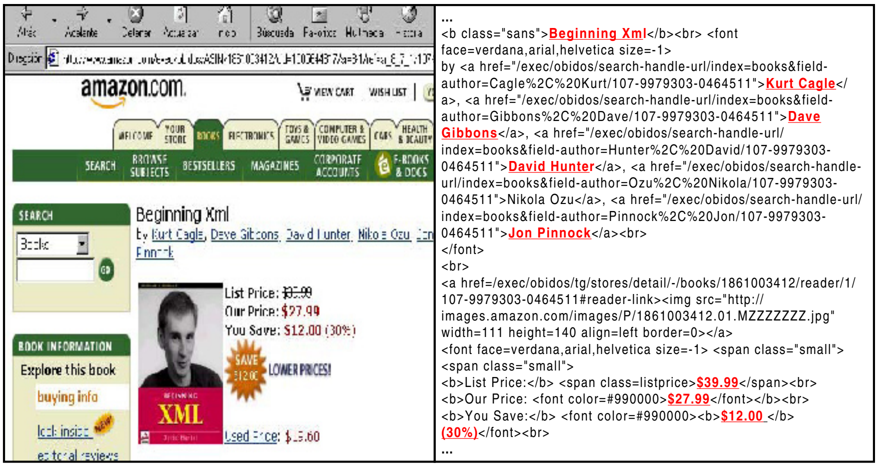
Thus, if we want to succeed in the development of web agents, we need a framework in which they can be separated from the information sources or the way to extract semantically-meaningful information from them. This way we enhance modularity, adaptability and maintainability, as well as agent interoperability. In this article, we present such a framework. It is organised as follows: Section 2 goes into details about our motivation and some related work; Section 3 presents our proposal and a case study; finally, Section 5 reports on our main conclusions and future research directions.

2 Motivation and Related Work

The incredible successfulness of the Internet world has paved the way for technologies whose goal is to enhance the way humans and computers interact on the web. Unfortunately, the information a human user can easy interpret is usually difficult to be extracted and interpreted by a web agent.

Figure 1 shows two views of a web page picked from `Amazon.com`. On the left, we present a shot of the page a human sees when he or she searches for information about a book; on the right, we present a portion of HTML that describes how to render this page. If we were interested in extracting the information automatically, the following issues would arise immediately:

– The implied meaning of the terms that appear in this page can be easily interpreted by humans, but there is not a reference to the ontology that describes them precisely, which complicates the communication interface between user and agent, and the interoperability amongst agents. For instance,



```
...
<b class="sans">Beginning Xml</b><br> <font
face=verdana,arial,helvetica size=-1>
by <a href="/exec/obidos/search-handle-url/index=books&field-
author=Cagle%2C%20Kurt/107-9979303-0464511">Kurt Cagle</
a>, <a href="/exec/obidos/search-handle-url/index=books&field-
author=Gibbons%2C%20Dave/107-9979303-0464511">Dave
Gibbons</a>, <a href="/exec/obidos/search-handle-url/
index=books&field-author=Hunter%2C%20David/107-9979303-
0464511">David Hunter</a>, <a href="/exec/obidos/search-handle-
url/index=books&field-author=Ozu%2C%20Nikola/107-9979303-
0464511">Nikola Ozu</a>, <a href="/exec/obidos/search-handle-url/
index=books&field-author=Pinnock%2C%20Jon/107-9979303-
0464511">Jon Pinnock</a><br>
</font>
<br>
<a href=/exec/obidos/tg/stores/detail/-/books/1861003412/reader/1/
107-9979303-0464511#reader-link><img src="http://
images.amazon.com/images/P/1861003412.01.MZZZZZZZ.jpg"
width=111 height=140 align=left border=0></a>
<font face=verdana,arial,helvetica size=-1> <span class="small">
<span class="small">
<b>List Price:</b> <span class=listprice>$39.99</span><br>
<b>Our Price: <font color=#990000>$27.99</font></b><br>
<b>You Save:</b> <font color=#990000><b>$12.00 </b>
(30%)</font><br>
...
```

Fig. 1. A web page that shows information about a book.

if we do not consider semantics and we interpret prices as real numbers, a US librarian agent that deals with dollars would be likely to misinterpret a European one that deals with euros.

– The layout and the aspect of a web page may change unexpectedly. For instance, it is usual that web sites incorporate Christmas stuff in December, which does not affect the information they provide, but may invalidate the automatic extraction methods used so far.
– The access to the page that contains the information in which we are interested may involve navigating through a series of intermediate pages, such as login or index pages. Furthermore, this information may spread over a set of pages.

On sight of these issues, several researchers began working on proposals whose goal is to achieve a clear separation between presentation concerns and data themselves. XML [12] is one of the most popular languages aimed at representing structured data, but, unfortunately, it lacks a standardised way to link them with an abstract description of their semantics. There are many proposals that aim at solving this problem. They usually rely on annotating web pages with instances of ontologies that is usually written in a language such as DAML [1] or SHOE [19].

Most authors agree in that it would be desirable for a web in which pages are annotated with semantics to exist, because this would help web agents understand their contents, and would enhance semantic interoperability. Unfortunately, there are very little annotated pages if we compare them with non-annotated pages. As of the time of writing this article, the DAML crawler (`www.daml.org/crawler`) reports 17,019 annotated web pages, which is a negligible figure if we compare it with 2,110 millions, which is the estimated number of web pages (`Cyveillance.com`). Current trends seem to suggest that the semantic web is not likely to be adopted in the forthcoming years [14]. This argues for an automatic solution to extract semantically-meaningful information from the web that is clearly separated from the business logic so as to enhance modularity, adaptability, and maintainability.

Several authors have worked on techniques for extracting information from the web, and inductive wrappers are amongst the most popular ones [17,23,15,4], [21,2]. They are components that use a number of extraction rules generated by means of automated learning techniques such as inductive logic programming, statistical methods, and inductive grammars. These techniques use a number of web pages as samples that feed an algorithm that uses induction to generalise a set of rules that allows to extract information from similar pages automatically. Recently, researchers have put a great deal of effort to deal with changes, so that extraction rules can be regenerated on the fly if the layout of the web page changes [18,16]. Although induction wrappers are suited to extract information from the web, they do not associate semantics with the extracted data, This being their major drawback.

There are also some related proposals in the field of databases, e.g., TSIMMIS [11] and ARANEUS [20]. Their goal is to integrate heterogeneous information sources such as traditional databases and web pages so that the user can work on them as if they were a homogeneous information source. However, these proposals lack a systematic way to extract information from the web because extraction rules need to be implemented manually, which makes them not scalable and unable to recover from unexpected changes on the web.

3 Our Proposal

Our proposal aims at providing agent developers with a framework in which they can have access to semantically-meaningful data that resides on web pages not annotated with semantics. It relies on using a number of agents that we call information channels or IC for short. They allow to separate the extraction of information from the logic of an agent, and offer agent developers a good degree of flexibility. In order to allow for semantic interoperability, the information they extract references a number of concepts in a given application domain that are described by means of ontologies.

Before going into details, it is important to say that our notion of agent was drawn from [24]: *"Agents have their own will (autonomy), they are able to interact with each other (social ability), they respond to stimulus (reactivity), and they take initiative (pro-activity)."* In our proposal, web agents are software agents that interact with the web to retrieve, extract or manage information.

3.1 The Architecture

Figure 2 sketches the architecture of our proposal. As we mentioned above, information channels are at its core agents because they specialise in extracting information from different sources, and are able to react to information inquiries (reactivity) from other agents (social ability). They act in the background proactively according to a predefined schedule to extract information and to maintain the extraction rules updated.

Each information channel uses several inductive wrappers to extract information so that they can detect inconsistencies amongst the data each one extracts. If such inconsistencies are found, they then use a voting algorithm to decide whether to use the data extracted by most wrappers or regenerate the set of extraction rules on the fly. This may happen because of an unexpected change to the structure of the web page that invalidates the extraction rules used so far.

There is also an agent broker for information extraction that acts as a trader between the agents that need information from the web and the set of available information channels. When an agent needs some information, it contacts the broker, which redirects the request to the appropriate information channel, if possible. This way, agents need not be aware of the existence of different ICs, which can thus be adapted, created or removed from the system transparently. However, every time an IC is created or destroyed, it must be registered or unregistered so that the broker knows it. It therefore has a catalogue with the description of every IC in the system (yellow pages).

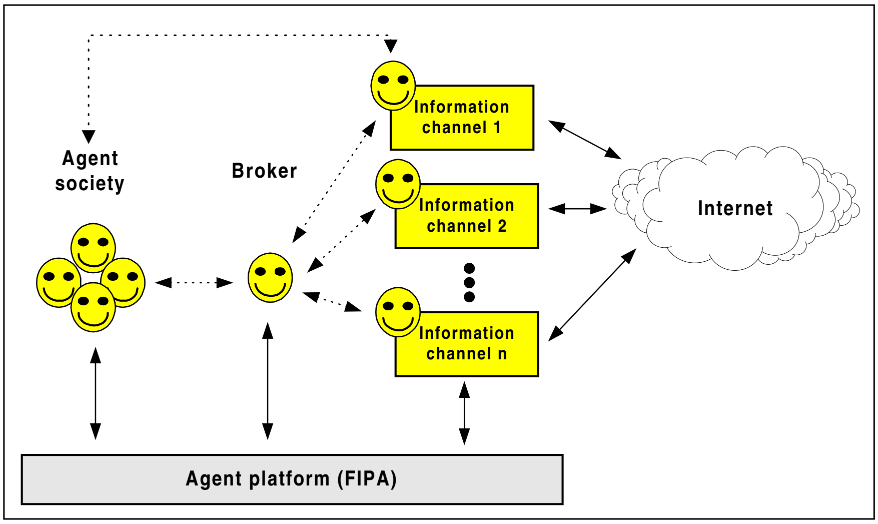


Fig. 2. Proposed architecture.

We use ACL [10] as a transport language to send messages from an agent to another. The content of the messages describes how an agent wants to interact with another, and it is written in DAML. Figure3 shows the brokering protocol [9] to communicate user agents with the ICs using the notation AUML [13,22]. When an initiator agent sends a message with the performative *proxy* to the broker, it then replies with one of the following standard messages: *not-understood*, *refuse* or *agree*. If the broker agrees on the inquiry, it then searches for an adequate IC to serve it. If not found, it then sends a *failure-no-match* message to the initiator; otherwise, it tries to contact the IC and passes the inquiry onto it. If the broker succeeds in communicating with the IC, this will later send the requested information to the initiator; otherwise, a *failure-com-IC message* is sent back to the initiator, which indicates that an appropriate IC exists, but cannot respond.

3.2 A Case Study

We illustrate our proposal by means of a simple example in which we are interested in extracting information about books from `Amazon.com`. We first need to define an information channel, i.e., we need to construct an expert system whose goal is to extract information about books from the web. Such a channel is characterised by the following features: a set of ontologies, a set of extraction rules, and a navigational document.

The Ontologies. The ontologies [6] associated with an IC describes the concepts that define the semantics associated with the information we are going

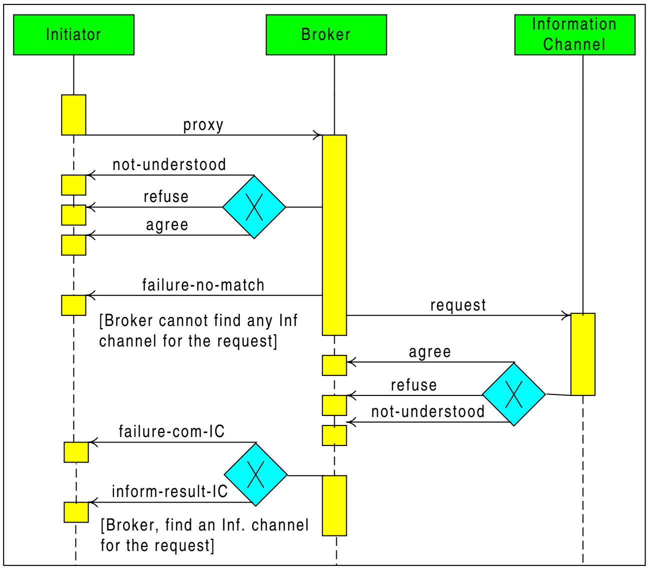


Fig. 3. Broker Interaction Protocol in AUML.

to extract from a high-level, abstract point of view. An ontology allows to define a common vocabulary by means of which different agents may interoperate semantically.

Figure 4 shows a part of the ontology that describes books using DAML. We selected this language because it is one of the most important contributions to the semantic web, it is being supported by many researchers, and the repository of available tools is quite rich, c.f. `http://www.daml.org/tools`.

The Extraction Rules. The extraction rules allow to define how to have access to the information in which we are interested. To generate them, we need a set of sample pages containing test data on which we use inductive techniques. To endow the sample pages with semantics, we also need to annotate them with DAML tags that allow to associate the concepts they contains with their corresponding ontologies. Figure 5 shows a piece of DAML code that we can use to annotate the web page in Fig. 1.

Once the sample pages have been annotated, we can generate the extraction rules. The input to the wrapper generator is a tuple of the following form:

$$(\{O_1, O_2, ..., O_n\}, \{(P_1, D_1), (P_2, D_2), ..., (P_k, D_k)\}); n \geq 1, k \geq 1$$

The first element of the tuple denotes the set of ontologies under consideration, and the second element is a set of pairs of the form (P, D), where P denotes a web page containing sample data, and D its corresponding annotation. With this information, we apply several induction algorithms [17,23,15,4] to generate

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
         xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#">

<daml:Ontology about="">
  <daml:versionInfo>10/30/2001</daml:versionInfo>
  <rdfs:comment>Book Ontology</rdfs:comment>
  <daml:imports rdf:resource="http://www.daml.org/2001/03/daml+oil"/>
</daml:Ontology>

<daml:Class rdf:ID="Book">
  <rdfs:comment>
    A set of written sheets of skin or paper or tablets of wood or
    ivory. Websters Dictionary.
  </rdfs:comment>
</daml:Class>

<daml:DatatypeProperty rdf:ID="isbn">
...

</rdf:RDF>
```

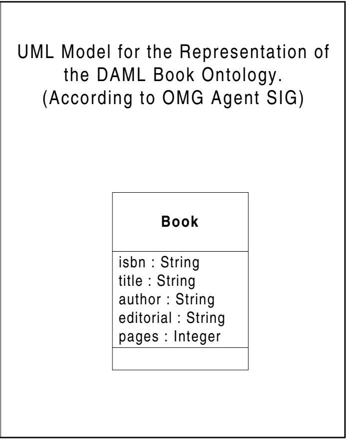

UML Model for the Representation of the DAML Book Ontology. (According to OMG Agent SIG)
Book
isbn : String
title : String
author : String
editorial : String
pages : Integer

Fig. 4. An ontology that describes books from an abstract point of view.

```
<Book rdf:ID="A book">
  <rdfs:label>A book</rdfs:label>
  <rdfs:comment>Instance of Book.</rdfs:comment>
  <title>Beginning Xml</title>
  <isbn>1861003412</isbn>
  <editorial>Wrox Press, Inc</editorial>
  <author>Kurt Cagle</author>
  <author>Dave Gibbons</author>
  <author>David Hunter</author>
  <author>Nikola Ozu</author>
  <author>Jon Pinnock</author>
  <pages>823</pages>
</Book>
```

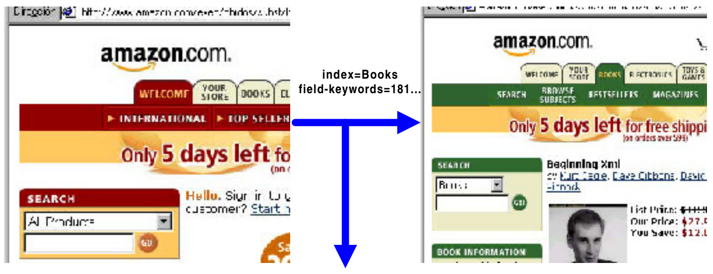
Fig. 5. Annotations to a web page with information about a book.

a set of extraction rules *R1, R2, ..., Rm*. Their exact form depend on the algorithm used to produce them, and may range from simple regular expressions to search procedures over a DOM Tree [5] or even XPointers [8]. hereafter, we refer to this set of rules as *BookRules*.

The Navigational Document. A navigational document deﬁnes the path of intermediate pages, such as login or index, we need to visit in order to have access to the page that contains the information in which we are interested.

Figure 6 shows a navigational document that describes how to have access to the web page that contains information about a book with a given ISBN. This document indicates that there exists a web page at `www.amazon.com` called d1. This page has quite a complex structure, but it contains a form called *search-form* with two ﬁelds a human user may use to select the kind of product he or she is searching for (*index*) and some key words that describe it (*field-keywords*).

Thus, d2 denotes the web page that we obtain when the server processes this form using *“index” := “Books”* and *“field-keywords” := “:Book#isbn”*. Books is the value that indicates the server that we are looking for books, and *:Book#isbn* denotes the ISBN code associated with the book for which we are searching.



```
Navigational document NDBook
    Using ontology http://arjona.lsi.us.es/Book#;

    Document d1:="http://www.amazon.com/";
    Document d2:=d1{searchform, "index" := "Books", "field-keywords" := :Book#isbn};

    Extract Info from d2 using BookRules;
End.
```

Fig. 6. A sample navigational document.

Roughly speaking, a navigational document may be viewed as a state machine in which each state is a web page, and the transitions between states may be viewed as the navigation from the page associated with the initial state to the page associated with the destination state.

3.3 Dealing with Pagers and Indexers

In this section, we present a slightly modified version of the previous case study in which we are interested in extracting information about the set of books written by a given author. If we write the name of an author in the above-mentioned search form, we can get a result page similar to the one presented in Fig. 1. In this case, the initial page does not lead directly to the requested information. Instead, it leads to an index page in which we can find some links that point to pages that describe books written by that author, and a pager that allows to search for more results. Although the wrapper we generated previously keeps working well, the way to have access to the information has changed substantially.

The new navigational document we need is presented in Fig. 7. There, we define a new document called *d3* that represents the web page to which the initial search form leads. *d4[]* references the set of web pages that we can get if we dereference the link labelled with *“More results”* by means of a sequential pager. This is an artefact that allows to navigate through the complete set of

```
Navigational document NDBook
   Using ontology http://arjona.lsi.us.es/Book#;

   Document d1 := "http://www.amazon.com/";
   Document d2 := d1{"searchform", "index" := "Books", "field-keywords" := :Book#isbn};
   Document d3 := d1{"searchform", "index" := "Books", "field-keywords" := :Book#author};
   Document d4[] := SequentialPager(d3, "More Results");
   Document d5[] := Indexer(d4[], AmazonIndexer);

   Extract Info from d2, d5[] using BookRules;
End.
```

Fig. 7. Modified version of the navigational document.

pages that contain the information about the books written by an author. *d5[]* references the set of web pages we obtain by dereferencing the links on books that appear in the set of pages *d4[]*. This set is obtained by means of an indexer, which is an artefact that analyses a set of pages and extracts the links to the pages in which we are interested. In this case, we use an inductive wrapper called *AmazonIndexer*, but new indexers can be easily generated.

3.4 Using the Information Channel

Once we have set up an abstract channel, we can send messages to it in order to retrieve information about a given book by means of the broker. The content of the messages in DAML is based on an ontology that deﬁnes the communication [7]. This ontology is illustrated in Fig. 8.

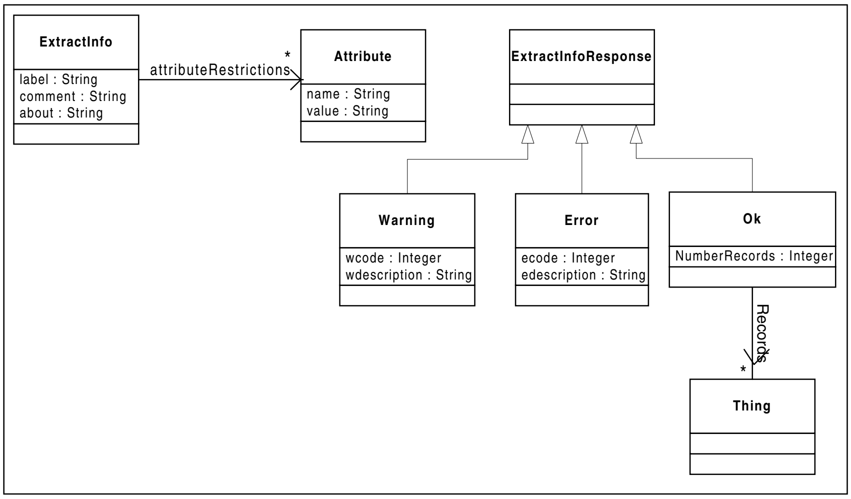


Fig. 8. Ontology for the content language.

Information requests are expressed as instances of class *ExtractInfo*. The reply from the abstract channel is an instance of *ExtractInfoResponse*, an error message (for instance, the channel can not have access the web page that contains the information), a warning message (for instance, 0 records have been found) or the information requested by the agent (as instances of the ontology class that defines the channel).

Figure 9 shows two example messages, the first one relative to an information request from an agent called *Agent-1* to the broker agent; the second one is the reply from the IC to *Agent-1*. In this case a book instance is returned.

```
(request :sender Agent-1
      :receiver broker
      :content (
         <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
               xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
               xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
               xmlns:xsd="http://www.w3.org/2000/10/XMLSchema#"
               xmlns:libro="http://arjona.lsi.us.es/Book#"
               xmlns="http://arjona.lsi.us.es/ie#">
         <ExtractInfo rdf:ID="EI1">
         <rdfs:label>EI1</rdfs:label>
         <about>http://arjona.lsi.us.es/Book#Book</about>
         <attributeRestrictions>
            <attribute>
                <name>http://arjona.lsi.us.es/Book#isbn</name>
                <value>1861003412</value>
            </attribute>
         </attributeRestrictions>
         </ExtractInfo>
         </rdf:RDF>
      )
      :language daml)
```

```
(inform :sender BookChannel
     :receiver Agent-1
     :content (
        <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
              xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
              xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
              xmlns:xsd="http://www.w3.org/2000/10/XMLSchema#"
              xmlns:ie="http://arjona.lsi.us.es/ie#">
              xmlns="http://arjona.lsi.us.es/Book#">
     <ie:Ok rdf:ID="Request 123332">
        <ie:numberRecords>1</ie:numberRecords>
        <ie:records>
           <Book rdf:ID="A-Book">
              <title>Beginning Xml</title>
              <isbn>1861003412</isbn>
              <editorial>Wrox Press Inc</editorial>
              <author>Kurt Cagle</author>
              <author>Dave Gibbons</author>
              <author>David Hunter</author>
              <author>Nikola Ozu</author>
              <author>Jon Pinnock</author>
              <pages>823</pages>
           </Book>
        </ie:records>
     </ie:Ok>
     )
     :language daml)
```

Fig. 9. Example of messages.

4 Benefits of the Framework for the Adaptative Hypermedia Community

The Adaptative Hypermedia Community can use the framework presented in this paper to personalise web sites based on users' profiles. In this respect, the information from the web that is interesting to a user can be extracted automatically and displayed in a suitable form.

We illustrate this idea with a fictitious case study whose aim is the personalisation of an e-commerce portal that sells VHS, DVDs, and so on. Using the information infered from the buys by some customers, we can identify who the the preferred actors are. Once the actors have been indentified, we can develop an agent that uses the proposed framework. It queries an abstract channel that is able to extract relevant information about this actor (birth name, location, filmography, ...) from several web sites, for example The Internet Movie Database (IMDb)[1].

[1] http://www.imdb.com/

5 Conclusions and Future Work

In this article, we have shown that the process of extracting information from the Internet can be separated from the business logic of a web agent by means of abstract channels. They rely on inductive wrappers and can analyse web pages to get information with its associated semantics automatically.

Our proposal shows that there is no need to annotate every web page, to extract information with associated semantics. In this sense we are contributing to bring together the community of agents programmers and the semantic web.

In the future, we are going to work on an implementation of our framework in which data sources can be more heterogeneous (databases, news servers, mail servers, and so on). Extraction of information from multimedia sources such as videos, images, or sound les will be also paid much attention.

References

1. DARPA (Defense Advanced Research Projects Agency). The darpa agent mark up language (daml). http://www.daml.org, 2000.
2. R. Baumgartner, S. Flesca, and G. Gottlob. Visual web information extraction with lixto. In *27th VLDB Conference*, 2001.
3. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
4. W. W. Cohen and L. S. Jensen. A structured wrapper induction system for extracting information from semi-structured documents. In *Workshop on Adaptive Text Extraction and Mining (IJCAI-2001)*, 2001.
5. W3C (The World Wide Web Consortium). Document object model. http://www.w3.org/DOM/, 2000.
6. O. Corcho and A. Gómez-Pérez. A road map on ontology specification languages. In *Workshop on Applications of Ontologies and Problem solving methods. 14th European Conference on Artificial Intelligence (ECAI'00)*, 2000.
7. S. Cranefield and M. Purvis. Generating ontology-specific content languages. In *Proceedings of Ontologies in Agent Systems Workshop (Agents 2001),*, pages 29–35, 2000.
8. S.J. DeRose. Xml linking. *ACM Computing Surveys*, 1999.
9. Finin, T. Labrou, and Y. Mayfield. Kqml as an agent communication language. *Software Agents, MIT Press*, 1997.
10. FIPA (The Fundation for Intelligent Physical Agents). Fipa specifications. http://www.fipa.org/specifications/index.html.
11. H. García-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. Integrating and accessing heterogeneous information sources in tsimmis. In *The AAAI Symposium on Information Gathering*, pages 61–64, March 1995.
12. C.F. Goldfarb and P. Prescod. *The XML Handbook.* Prentice-Hall, 2^{nd} edition, 2000.
13. OMG (Object Management Group). Unified modelling language version 2.0. http://www.omg.org/uml/, 2001.
14. J. Hendler. Agents and the semantic web. *IEEE Intelligent Systems Journal*, 2001.
15. C. A. Knoblock. Accurately and reliably extracting data from the web: A machine learning approach. *Bulletin of the IEEE Computer Society Technical Com-mittee on Data Engineering*, 2000.

16. N. Kushmerick. Regression testing for wrapper maintenance. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-1999)*, pages 74–79, 1999.
17. N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(2000):15–68, 1999.
18. N. Kushmerick. Wrapper verification. *World Wide Web Journal*, 2000.
19. S. Luke, L. Spector, D. Rager, and J. Hendler. Ontology-based web agents. In *First International Conference on Autonomous Agents*, 1997.
20. G. Mecca, P. Merialdo, and P. Atzeni. Araneus in the era of xml. *Data Engineering Bullettin, Special Issue on XML*, September 1999.
21. I. Muslea, S. Minton, and C. Knoblock. Wrapper induction for semistructured, web-based information sources. In *Proceedings of the Conference on Automated Learning and Discovery (CONALD)*, 1998.
22. J. Odell, H. Van Dyke, and B. Bauer. Extending uml for agents. In *AOIS Worshop (AAAI)*, pages 3–17, 2000.
23. S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, pages 1–44, 1999.
24. M. J. Wooldridge and M. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.