Utilization of UML diagrams in designing an events extraction system

MIHAI AVORNICULUI

Babes-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania
mavornicului@yahoo.com

Keywords: event extraction, RUP method, UML diagram

The system proposed and investigated in this paper is intended to be a helpful instrument when looking for pieces of information on different web pages, in cases when a special type of event is needed to be centralized. The system can be useful, for example, when information about weather forecast for a specific geographical region is needed to be collected from different web pages (Yahoo, Google, CNN site, etc.). To develop such a system we will use the UML diagrams.

1. Introduction

Nowadays the amount of information on the Internet reaches high proportion. In finding specific information on the Internet some general instruments (engines of search) have become popular, which automatically run through all the existent web pages in order to update the databases containing the latest information on the Internet. In most cases the search is done based on a number of strings stored in the database of the search engine. The result of such a search is, generally, a large amount of links to different web pages.

To systematize the searching process and to obtain a result in a concrete form, an other stage is useful, a stage in which the information returned by the search engine is processed, and the response is generated in a more organized form [3,5,6].

The centralization of a specific type of event is useful, first of all, to realize some news services. These services must offer updated information about a specific type of event, and – if possible – in real time.

Take sport events for example. A user of this type of news service might want to find out what sport events take place in a certain geographical region (a city, a country, etc.) during a certain period of time (in that very moment, a day before, or the next week, etc.). All this information has to be obtained from centralized information. In this way, useful data can be obtained about a specific event from various sources. These sources (the websites from where the information was taken) can complete each other concerning the information content about a specific event.

The system will recognize the events of a specific type (weather, sport, politics, and text data mining) depending on the way it will be drifted (the dictionary that it possesses). These events can be transmitted to the user or the entire context in which the event appeared, and can be taken, to show the initial form in which the event was included.

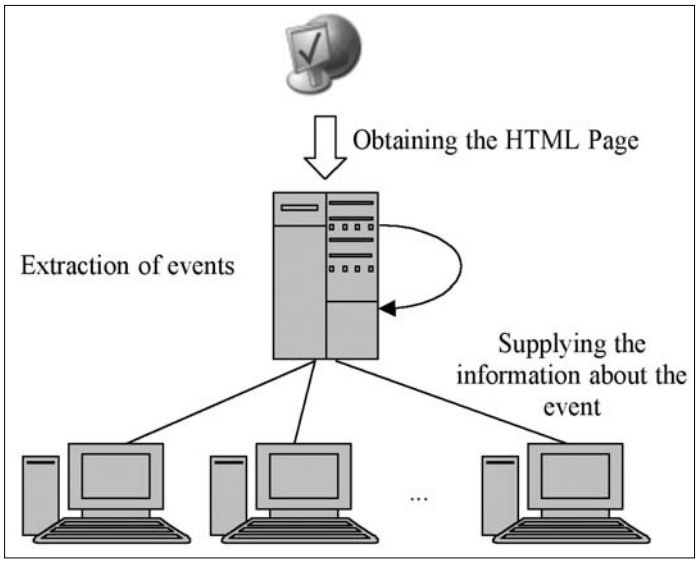
2. The conceptual model

To be able to realize what we intended, the system (the components) has to handle, in general, three aspects [1]:

• Taking the HTML documents from the Internet and saving them in a database in order to process them later: in order to realize this aspect, the system must gather and run through a number of webpage addresses. This list of addresses can depend on type of the event we want to identify. The application will recursively run through the list of addresses and will save all the documents it meets in the local database.

• Document processing and obtaining the information needed: the processing of documents and the extraction of the information needed is based on a dictionary of concepts which describes the types of events. This dictionary of concepts has to be flexible in order to be able to identify different variations of types of events. All the identified events will be stored in a database in order to be able to refer to them later.

Figure 1. The structure of the system



• Giving the users a way of access to the collected information: finally, an access to the extracted information has to be given to the user. For example, in case of sport events, a list of arranged sport events can be presented to the users, according to the date when it will take place or when it took place. The search on different criteria will be allowed. This offers a quick access to the information requested, eliminating the necessity of day-by-day search.

The *Figure 1* relevantly presents the structure of the system.

In case of human users, HTML documents will be offered with the needed information, and if the user is a computer programme, then the information will be transmitted using a generic form, for example an XML document, in order to be able to process it easily.

3. The application of the RUP method

RUP (Rational Unified Process) is an iterative method. Each iteration has one or more aims. In RUP, the aim is to produce a functional software which can add value and deliver it to the customer. The iterations are determined by a time limit. This means that each facility must be realised in a certain period of time [8,10,11].

According to Kruchten, RUP has the following characteristics [4]:

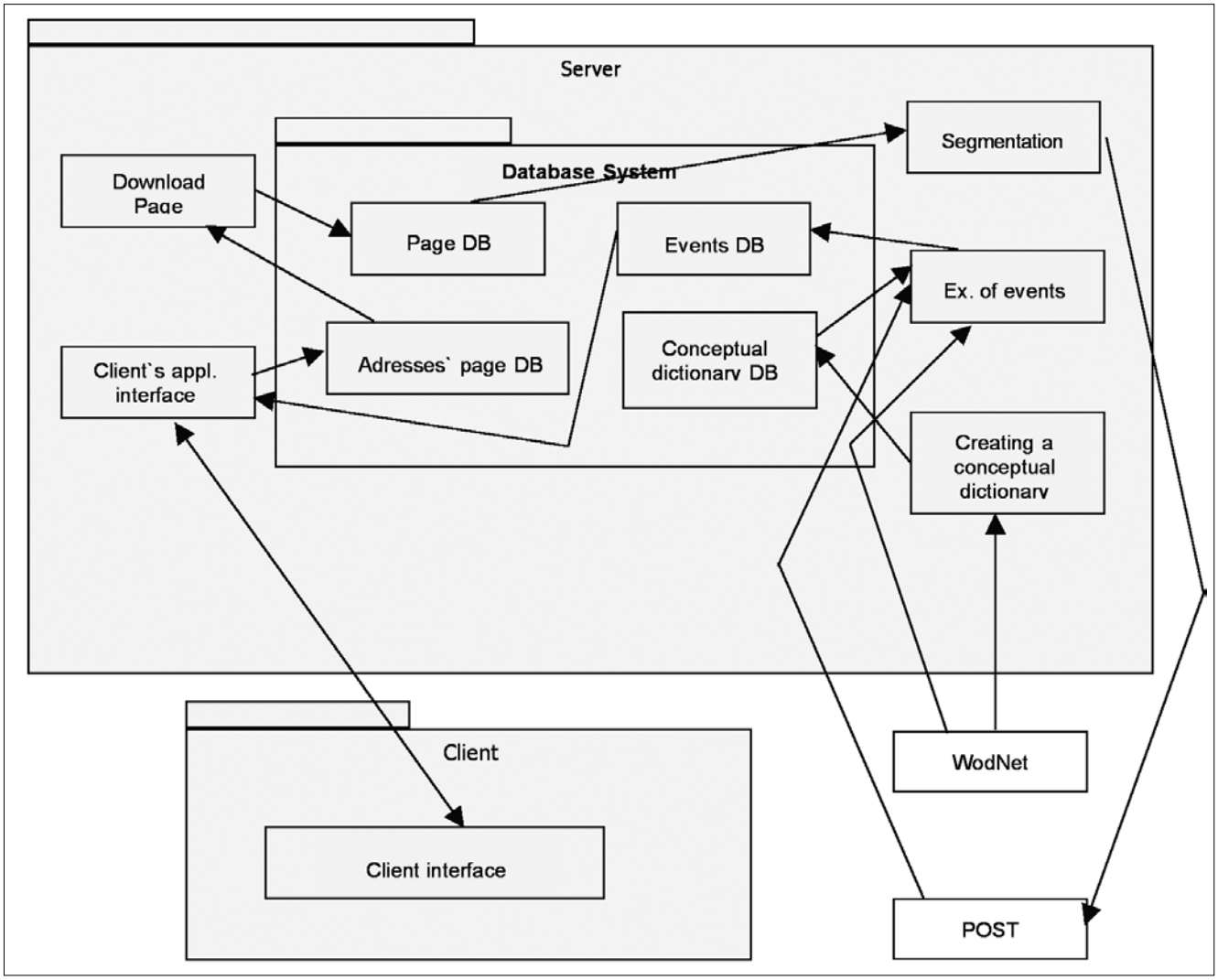
• *Interactive development of a software product* – proposes developing in short increments of certain iteration chains. This ensures a real-time detection of risks and an adequate addressing of them.

• *Processing demands* – denotes a continuous process of demand identification of a system that evolves in time and the demand factors that have the greatest impact on the system. Processing these demands requires a disciplined manner of evaluating, associating priorities and monitoring. It is better that communication had a well defined set of demands as a base.

• *Uses architecture based on components* – it is more flexible, making the extensibility of the particular application possible. Components can be redesigned or extended without compromising on the evolution of the whole system.

• *Uses visual instruments of modelling contributing to the understanding of extremely complicated systems*

Figure 2. The diagram of the modules



Module	Function
Addresses' page Database	Contains the list of web pages for a certain domain, on which the search will take place.
Events Database	Contains the identified events, indexed according to their domain, time, locations and criteria.
Conceptual dictionary Database	Contains the concepts which are going to be looked for, indexed according to domains.
Download Page	Harvests periodically web pages from the Internet with addresses in the page addresses Database, and stocks them in Pages Database. This module can recursively fetch the pages indicated by the links contained by the current page, from the same server, to a certain depth of harvesting.
Segmentation	Divides the text from the web pages of the pages database in segments.
POST	This module has the role to process the segments and to annotate them with the adequate parts of speech from the segment.
Extraction of events	Extracts the events from the annotated segments from the Post module, on account of the concepts from the conceptual dictionary and from the WorldNet, depositing them into the events database.
Creating the conceptual dictionary	Constructs the specific concepts of a certain domain through a learning algorithm, and after that deposits them into the Conceptual database.
WordNet	Supplies relations between words.
Client's application interface	Gets requests of looking for events from the Events database.
Client's interface	Assures the interface with the client, the introduction of events' queries, options set ups, displaying the results according to several criteria.

Table 1. The functions of the modules

– using UML models the complexity of a system can be effectively processed among more developers.

• *Permanently verifies the quality of the produced software* – this represents a constant occupation that runs at the level of every iteration. From this perspective errors are discovered in time and revising costs are reduced.

• *Controls changes brought by developed software* – dealing with these changes represents a key to the success of developing an IT system. If one of the team members causes a change to the system, every member has to be warned, who is affected by that change.

Analysing the characteristics of the RUP method, we can conclude the following regarding the advantages of the method when it is used to develop systems of event extractions [1]:

• RUP is a method that enables developing systems with a flexible and extendible architecture. Event extraction systems are these kinds of systems so they comply with these demands.

• RUP focuses on dealing with aspects of potential risk in time. This characteristic is a plus for every system.

• It doesn't imply a fixed set of tasks in the initial stage so they can be refined as the project evolves. From the point of view of an event extraction system tasks can not be specified in the first stages so this characteristic is in favour of event extraction systems.

• RUP stresses on the final product and on the conformity of this with the demands of the final users. This is clearly an advantage of event extraction systems.

• RUP leaves the evolution of the system entirely on the users will. From the point of view of developing an event extraction system, this characteristic can turn into an important disadvantage. It is possible that the user does not have any knowledge of EDI or XML message formats so he could jeopardize the flexibility of the system.

• RUP takes over the advantages offered by UML. For a lot of IT systems this represents an advantage.

• RUP makes possible to control the quality of the developed system. The quality of the system is in close relationship with its reliability. The better the quality the more reliable the system is. This characteristic brings an advantage to developing an event extraction system.

• The time needed to develop a system using RUP is much less than in case of other methods, so this is also considered an advantage.

• When it is adopted RUP becomes a repetitive and predictable process for the developing team. This leads to a high efficiency in case of developing large and reliable software.

We consider that RUP represents a method that can be successfully used for developing event extraction systems while avoiding certain disadvantages of the method. Among these is the fact that RUP does not contribute in an explicit way to the development of some implementing instructions, for it is a perspective method in comparison with the more agile XP.

4. The modules of the system

Just like in the case of other systems, this system is also composed of modules. Now we will identify the system's modules *(Figure 2)*.

In *Table 1* the functions of each module is presented.

5. Detailed design of the system

The system will naturally contain several classes. The classes are not isolated elements of the system. Consequently, we will have to identify the relations between classes.

In the followings we will present the attributes and methods of the classes.

The Addresses' page DB class:

- Is an XML database.
- Contains the list of starting pages' addresses.
- In this database there are only the starting pages, but their harvesting module will search and fetch the pages that were referred to in the starting pages, from the same server, for a certain depth of searching

Example:

```
<xml>
    <domain name="weather">
        <url name="www.yahoo.com/weather" depth=3>
        <url name="www.cnn.com/weather" depth=2>
        .....
    </domain>
</xml>
```

The Pages Database:

- Is an XML database.
- Contains in XML format the indexes of brought pages, and, in separate files, the pages.
- Only the pages containing text are downloaded and kept (extract pieces of information).
- For each page there is an index in the page' s database which contains information about the URL of the page, the last update time, the name of the file where it is stored, a unique identifier for each page and the list of the page's identifiers to which the page is related to.

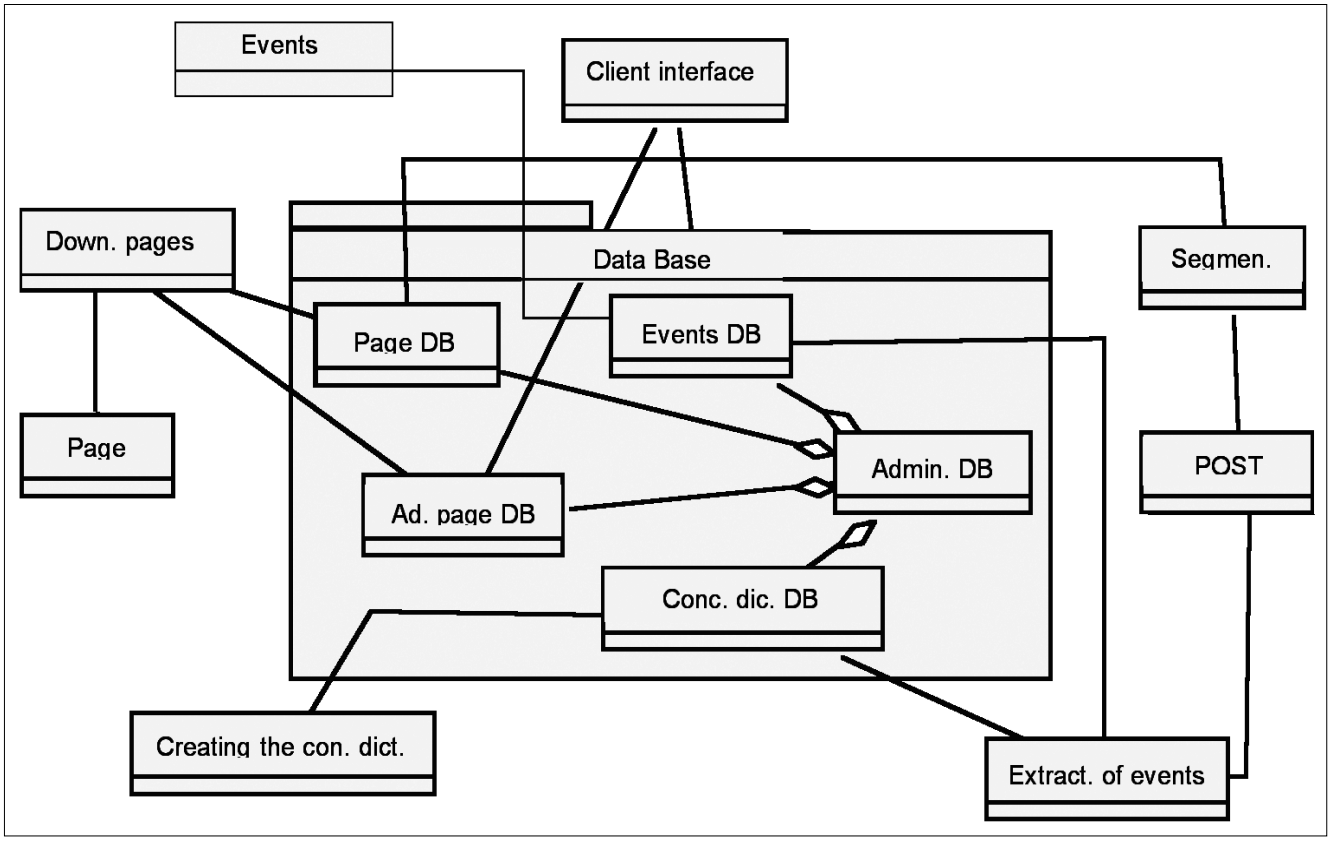
Example:

```
<xml>
    <domain name="weather">
        <URL page="www.yahoo.com/weather/"
        lastUpdated="22.10.2008"
        fileName="weather_com_yahoo_www" id=1>
        </page>
        <URL page="www.yahoo.com/weather/europe"
        lastUpdated="22.10.2008"
        fileName=
        "europe_weather_com_yahoo_www" id=2>
        <upPage>1</upPage>
        </page>
    </domain>
<xml>
```

The Events database class:

- Is an XML database.
- Keeps the identified events, indexed by domain, time, location and other criteria.
- An event contains the name of the event and a number of attributes specific to the event, ex. {**rain**, Cluj-Napoca, today}.

Figure 3. Relations between classes



The Conceptual dictionary database class:
- Is an XML database.
- Contains the concepts that are going to be looked for, indexed by domain.
- Concept – the collection of all information (events) of a certain type, which respects the set of syntaxes imposed by the concept ex: {**rain**, Madrid, today}, {**rain**, Bucharest, tomorrow}, {**rain**, Sofia, yesterday}, {**rain**, X, Y}, for each valid X and Y, plus a series of valid syntaxes for the rain concept and attributes (complements) of X and Y.
- For one concept the following information must be kept:
  – the name of the concept- releaser of the concept
  – type of concept
  – the list of attributes
  – the syntax of the concept – the relative position of the attributes to the name of the concept

The Page download class:
- Fetches periodically pages from the Internet from the addresses contained by the addresses' page database. This module can recursively fetch the pages indicated in the current page, from the same server, to a certain depth of harvesting.
- This module uses an external application, specialised in fetching recursively web pages, but it can apply the function internally.
- Has page analysing function to extract new links which are going to be used, plus an elimination function, which erases information (ex: scripts) which are not relevant for the application.

The Segmentation class:
- Divides the text of the web pages from their database into different segments.
- A segment is a part of a text, which can be a sentence or a complex sentence and which can be seen as an entity of atomic information.
- The post applications accept this kind of segments as input data.
- The segmentation is done on account of the HTML tags which help to delimit them (ex: <P>
 etc.) but also on the account of the text delimiters.

The POST class:
- This module has the role to process the segments and to annotate them with the adequate part of speech from the segment.
- The module interfaces the programme with a Part of Speech Tagger application, which receives a segment and annotates it.

Event Class:
- It contains the most important attributes of events.
- Interacts with the Events database.

The Extraction of events class:
- Extracts the events from the annotated segments from the Post modules, on account of the concepts from the conceptual dictionary and from the WordNet, depositing them into the events database.
- This module is one of the most complex modules of the system, together with the Creating the conceptual dictionary module and the segmentation module.
- A concept is formed of a trigger and a series of attributes, which can be located relatively to the trigger in a certain schema (which implies a certain syntax).
- Uses a pattern adjusting algorithm to identify the possible attributes, which are checked later using the pieces of information from the WordNet.

Admin DB class:
- It is an XML data base
- It holds necessary information to the data bases: Events, Page, Ad. Page and Conc. Dic.

Creating the conceptual dictionary class:
- Constructs the specific concepts of a certain domain through a learning algorithm, and then deposits them into the Conceptual database.
- The construction of the conceptual dictionary can be done either by identifying manually the representative concepts for a certain domain, which are provided to the module to enter them into the database; or by using the training and learning algorithm of certain concepts on special learning pages.
- The model of the learning algorithm leads to the extraction of concepts which match well and which can well identify the events from similar pages to those from which the learning has been done.
- The construction of dictionary is one of the most important components in this stage in extracting the events.

Client's application interface class:
- Gets the requests for looking for events, which are provided by the Events database.
- The events are only taken out from the database, their search and identification being made separately by the Events Extraction module.

In order to implement the system we can use Java Server Pages (JSP). JSP is the most popular method to create Web interfaces for the applications which are Java based.

6. Conclusions

Using UML in the development of event extraction systems is opportune from several points of view. UML offers powerful tools of modelling behaviours aspects. Classes contain both data and their associated processes. It also offers a complete vision above the groupings of different components under the form of packets and their physical places. Event extraction systems are complex systems that present a more dynamic evolution than other types of IT systems.

For this reason it is necessary to use a flexible instrument of analysis and design that permits the future expansion of the system.

Author

AVORNICULUI MIHAI-CONSTANTIN graduated from the Babes–Bolyai University of Sciences and obtained his M.Sc. degree in Databases and Electronic Commerce in 2005. He obtained a five-months SOCRATES/ERASMUS scholarship in 2004 which he spent at Johannes Kepler University in Linz. Since 2005 he has been responsible for the specialization of informatics in economics at Babes–Bolyai University of Sciences. He has been author or co-author of several lecture notes, textbooks and monographs during 2002-2007. Main research areas include databases, object oriented programming and modeling. Mr. Avornicului is currently is working toward his Ph.D. degree.

References

[1] Avornicului, C., Avornicului, M., The Use Of Objective Methods for Developing Events Extraction Systems, Annals of the Tiberiu Popovici Seminar, Cluj-Napoca, October 10-12, 2008., pp.11–22.

[2] Avornicului M., Planning and management of information systems, ÁBEL Publishers, Cluj-Napoca, 2007 (in Hungarian).

[3] Han, J. and Kamber, M., Data Mining: Second Edition Concepts and Techniques. Morgan Kaufman Publishers, 2006.

[4] Kruchten, P.B., The Rational Unified Process: An Introduction – IEEE Software, 1998.

[5] Lin, B., Web Data Mining – Exploring Hyperlinks, Contents an Usage Data, Springer, 2007.

[6] Lin, T.Y., Xie, Y., Wasilewska, A., Lian, C.J., Data mining: Foundations and Practice, Springer Berlin, 2008.

[7] Markov, Z., Larose, D.T., Data Mining the Web, John Wiley & Sons, 2007.

[8] Raffai M., The UML 2 modeling language, Palatia Printers and Publishers, 2005 (in Hungarian).

[9] Sieg, A., Mobasher, B., Burke, R., Ontological User Profiles for Personalized Web Search. Proceedings of AAAI Workshop on Intelligent Techniques for Web Personalization, AAAI Press Technical Report WS-07-08, July 2007., pp.84–91.

[10] Sommerville, I., Software Engineering, 7th Edition, Addison-Wesley, 2004.

[11] http://rup.hops-fp6.org/process/ovu_proc.htm