MONITORING WEB DATA SOURCES USING TEMPORAL PROPERTIES AS AN EXTERNAL RESOURCES OF A DATA WAREHOUSE

Francisco Araque, Alberto Salguero and Cecilia Delgado
Department of Software Engineering
E.T.S.I.I.T., University of Granada, Granada (Andalucía), Spain

Keywords: Data Warehouse, Temporal integration, Middleware Integration, Organisational Issues on Systems Integration.

Abstract: Flexibility to react on rapidly changing general conditions of the environment has become a key factor for economic success of any company, and the WWW has become an important resource of information for this proposal. Nowadays most of the important enterprise has incorporated the Data Warehouse (DW) technology where the information retrieved from different sources, including the WWW, has been integrated. The quality of data provided to the decision makers depends on the capability of the DW system to convey in a reasonable time, from the sources to the data marts, the changes made at the data sources. If we use the data arrival properties of such underlying information sources, the DW Administrator can derive more appropriate rules and check the consistency of user requirements more accurately. In this paper we present an algorithm for data integration depending on the temporal characteristics of the data sources and an architecture for monitoring web sources on the WWW in order to obtain its temporal properties. In addition we show an example applied to tourism area where data integrated into DW can be used to schedule personalized travel as a value-added service for electronic commerce.

1 INTRODUCTION

There is an increase in the number of data sources that can be queried across the WWW. Such sources typically support HTML forms-based interfaces and search engines query collections of suitably indexed data. One drawback to these sources is that there is no standard programming interface suitable for applications to submit queries. Second, the output is not well structured. Structured objects have to be extracted from the HTML documents which contain irrelevant data and which may be volatile. Third, domain knowledge about the data source is also embedded in HTML documents and must be extracted.

On the other hand, web information sources have their own information delivery schedules (Watanabe *et al.*, 2001). In such cases, data arrival time is predetermined or predictable. If we use the data arrival properties of such underlying information sources, the system can derive more appropriate rules and check the consistency of user requirements more accurately. If it is not available information about how data sources evolve throughout time, we uses different approaches to maintain temporal coherency of data gathered from web sources (Araque *et al.*, 2003).

Usually, the enterprise develops systems that are continuously polling the sources to enable (near) real-time changes capturing and loading. This approach is not efficient and can produce overload problems if it is necessary to query a lot of sources. It is more efficient to poll the web site when it is needed. Every web source evolves autonomously and it is necessary to check its evolution along time. In the context of the DW, the state of the environment as perceived by the controlling system (DW refreshment process) must be consistent with the actual state of the environment being controlled (value of the data in the data source).

In order to address the above problem, we propose a system to allow distributed information monitoring of web data sources on the WWW. The approach relies on monitoring information distributed on different resources and alerting the

Araque F., Salguero A. and Delgado C. (2007).
MONITORING WEB DATA SOURCES USING TEMPORAL PROPERTIES AS AN EXTERNAL RESOURCES OF A DATA WAREHOUSE.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - DISI*, pages 28-35
DOI: 10.5220/0002393100280035

user (in our case the DW refreshments process) when certain conditions regarding this information are satisfied (temporal properties). We shall also propose an algorithm for data integration depending on the temporal properties of the data sources.

Section 2 reviews the concepts of the DW and Data Sources. Section 3 introduces temporal properties extraction. Section 4 describes our proposal for data integration and the algorithm proposed. Section 5 shows a motivation example. And we finish with the conclusions.

2 DATA WAREHOUSE AND DATA SOURCES

Inmon (Inmon, 2002) defined a Data Warehouse (DW) as "a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision-making process." A DW is a database that stores a copy of operational data with an optimized structure for query and analysis. The scope is one of the issues which defines the DW: it is the entire enterprise. In terms of a more limited scope, a new concept is defined: a data mart is a highly focused DW covering a single department or subject area. The DW and data marts are usually implemented using relational databases (Hammer et al. 1995) which define multidimensional structures.

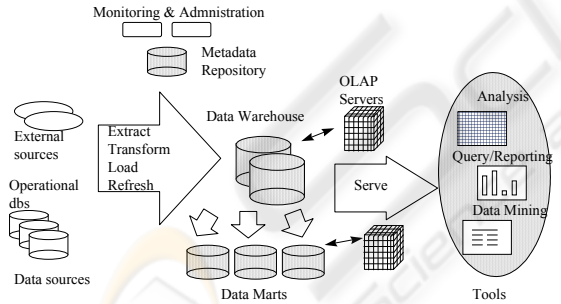

Figure 1: A generic DW architecture.

The generic architecture of a DW is illustrated in figure 1 (Chaudhuri & Dayal, 1997), which shows that data sources include existing operational databases and flat files (i.e. spreadsheets or text files) combined with external databases. Data is extracted from the sources and then loaded into the DW using various data loaders and ETL (Extraction, transformation and loading) tools (Araque, 2003a), (Araque, 2003b). The warehouse is then used to populate the various subject/process-oriented data marts and OLAP servers. Data marts are subsets of a DW which has been categorized according to functional areas depending on the domain (addressing the problem area) and OLAP servers are software tools that help a user to prepare data for analysis, query processing, reporting and data mining. The entire DW then forms an integrated system that can support various reporting and analysis requirements of the decision-making function (Chaudhuri & Dayal, 1997).

After the initial loading, warehouse data must be regularly refreshed, and modifications of operational data since the last DW refreshment must be propagated into the warehouse so that the warehouse data reflects the state of the underlying operational systems (Araque & Samos, 2003), (Araque, 2003b).

Data sources can be operational databases, historical data (usually archived on tapes), external data (for example, from market research companies or from the Internet), or information from the already existing data warehouse environment. They can also be relational databases from the line of business applications. In addition, they can reside on many different platforms and can contain structured information (such as tables or spreadsheets) or unstructured information (such as plain text files or pictures and other multimedia information).

Extraction, transformation and loading (ETL) are data warehousing processes which involve extracting data from external sources, adapting it to business needs, and ultimately loading it into the data warehouse. ETL is important as this is the way data actually gets loaded into the warehouse.

The first part of an ETL process is to extract the data from the source systems. Most data warehousing projects consolidate data from different source systems. Each separate system may also use a different data organization/format. Common data source formats are relational databases and flat files, but there are other source formats. Extraction converts the data into records and columns.

The transformation phase applies a series of rules or functions to the extracted data in order to derive the data to be loaded.

During the load phase, data is loaded into the data warehouse. Depending on the organization's requirements, this process can vary greatly: some data warehouses merely overwrite old information with new data; more complex systems can maintain a history and audit trail of all the changes to the data.

DWs describe the evolving history of an organization, and timestamps allow temporal data to be maintained. When considering temporal data for DWs, we need to understand how time is reflected in a data source, how this relates to the structure of the

data, and how a state change affects existing data. A number of approaches have been explored (Bruckner & Tjoa, 02).

Some specific architectures have been presented in order to extract data from Web accessible sources (Yu et al., 2006), but the authors mainly focus on the automatically data extracting methods from unstructured documents issues and do not pay adequate attention to the temporal aspects of the process.

3 TEMPORAL PROPERTIES EXTRACTION

In the information extraction process it is necessary to maintain the temporal consistency (Araque *et al.*, 2003a), (Srinivasan *et al.*, 98) of the data that the user needs in the sense put forward in real-time applications. In real-time applications, the state of the environment as perceived by the controlling system must be consistent with the actual state of the environment being controlled (Ramamritham *et al.*, 96). In this case we employ algorithms used to maintain coherency between a data source and cached copies of the data (Srinivasan *et al.*, 98). These algorithms try to minimize the number of times that the client has to connect to the server. Besides, we use this technique to maintain the temporal consistency between the data extracted from web information sources and the data loaded in the DW according to DW designer temporal requirements. Then, the DW Administrator can use the temporal properties of the data to adjust the refreshment parameters.

To solve these problems, we presented tools to define and generate wrappers for Web accessible sources (Araque, 2003a, 2003b). The toolkit provides a graphical interface to specify the capabilities of the sources and to define a simple mapping translation from data at the source level to data at the user level. Also, we can define when we want to extract data from the source according to temporal constraints specified by the user. We use *DETC* (Data Extraction with Temporal Constraints) to refer to the software tool (out of scope of this paper). We use the algorithms presented in (Srinivasan *et al.*, 98) and a wrapper generation tool to find out the temporal properties of a data source.

3.1 Temporal Properties of Data

The DW must be updated periodically in order to reflect source data updates. The operational source systems collect data from real-world events captured by computer systems (Bruckner *et al.*, 2002). The observation of these real-world events is characterized by a delay. This so-called propagation delay is the time interval it takes for a monitoring (operational) system to realize an occurred state change. The update patterns (daily, weekly, etc.) for DWs and the data integration process (ETL) result in increased propagation delays.

Having the necessary information available on time means that we can tolerate some delay (be it seconds, minutes, or even hours) between the time of the origin transaction (or event) and the time when the changes are reflected in the warehouse environment. This delay (or latency) is the overall time between the initial creation of the data and its population into the DW, and is the sum of the latencies for the individual steps in the process flow.

In this work, we consider the following temporal parameters (a more detailed explanation can be found in (Araque *et al.*, 2006b) to be of interest on the basis of the characteristics of the data extraction methods and the data sources:

– *VTstart*: time instant when the data element changes in the real world (event). At this moment, its Valid Time begins. The end of the VT can be approximated in different ways which will depend on the source type.

– *Granularity (Gr)*: It is the extent to which a system contains discrete components of ever-smaller size. Generally speaking, information granules are collection of entities, usually originating at the numeric level, that are arranged together due to their similarity, functional adjacency, coherency or the like. In our case, because we are dealing with time, it is common to work with granules like minute, day…

– *AW(Availability Window)*: Period of time in which the data source can be accessed by the monitoring programs responsible for data source extraction. There may be more than one daily availability window.

– *M:* time instant when the data source monitoring process is initiated.

By considering the previous temporal properties and two data sources with their specific extraction methods (this can be the same method for both), we can determine whether it will be possible to integrate data from two sources (according to DW Administrator requirements).

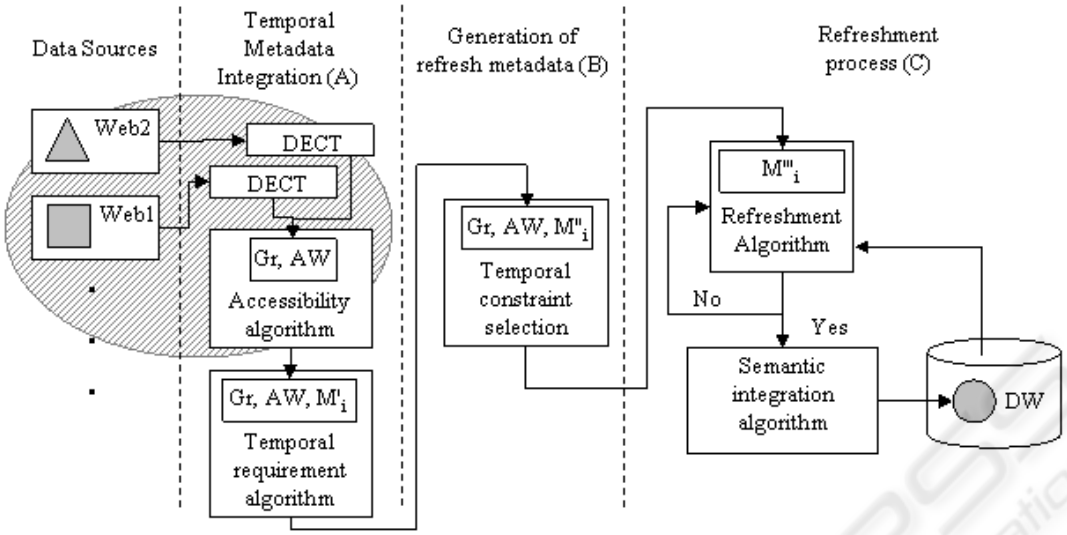

Figure 2: System architecture.

We can represent the temporal characteristics of the data source with these temporal concepts. It is therefore possible to determine *when* the data source can offer the data and *how* this data changes over time. This can be represented in the temporal component schema (Araque *et al.*, 1999) and used by the DW administrator to decide how to schedule the refreshment activity.

4 DATA INTEGRATION

Prior to integration, it is necessary to determine under what parameters it is possible and suitable to access the sources in search of changes, according to their availability and granularity, obtained automatically by the tool of the previous section. This process is carried out by the pre-integration algorithm. It is only possible to determine these parameters previously if there is some pattern related to the source availability. The parameters obtained as a result shall be used in the specific integration algorithms whenever the data sources are refreshed.

One of the most complex issues of the integration interface is the case where there are multiple sources for a single element of data in the DW. For example, in the DW there is a data element that has as its source data element a1 from legacy application A and a data element b1 from legacy application B. If it is possible to temporally integrate the data from both sources (on the basis of their temporal properties), semantic integration is undertaken and the result is stored in the DW.

The integration methodology, shown in figure 2, consists of a set of processes that define the rules for capturing a parameter from a single source as well as integrate a set of values semantically equivalent coming from different data sources. It has two phases, shown in figure 2: *Temporal integration* (A) and *Generation of Refresh metadata* (B). The elements of the architecture that are of interest in this paper have been shadowed in figure 2.

The temporary process of integration can also be divided into two different tasks: the analysis of the accessibility of both sources and the analysis of temporal requirements. The former task, which this article is focused in, verifies that certain temporary parameters common to any type of extraction method are satisfied, so the integration can be carried out, whereas the second one, which would be carried out only in the case of surpassing the first task, is focused on determining whether the integration of specific sources of data is possible. We obtain as result rules about the integration possibilities existing between the sources (minimum granularity which can be obtained, the intervals in which refreshment should be performed, etc). The second task, which has been widely described in (Araque *et al.*, 2006b), is not going to be explained more in depth because it uses some temporal properties that is out of the scope of this paper.

In the second phase the most suitable parameters are selected to carry out the refreshment process of the data. It is in this second phase where, from the minimum requirements selected by the temporary first stage of integration, the DW designer sets the refreshment parameters. These parameters can be set

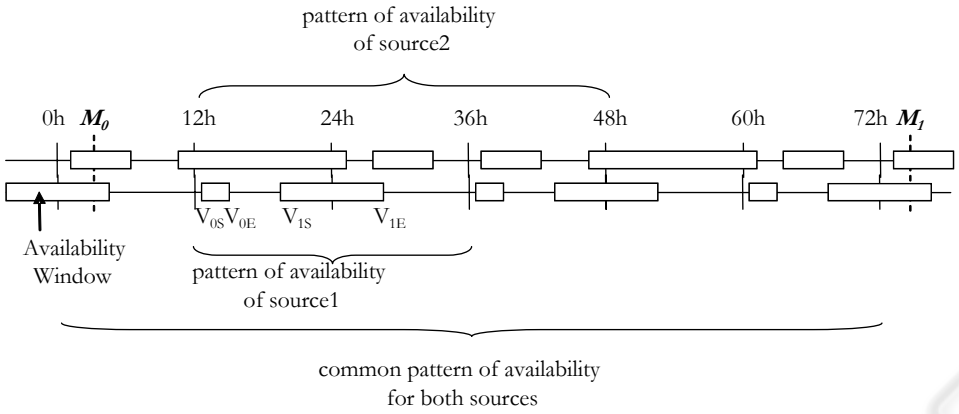

Figure 3: "*Common pattern of availability*".

```
In:
  source[] : list of sources that contains the semantically equivalent parameter to
             integrate
  commonAW : common Availability Window pattern.
Out:
  M[] : list of instants to query the sources

If commonAW is periodical then
  GrMax = MinDetail(Granularity(source[1]), Granularity(source[2]), …)
  // Example: day = MinDetail(hour, day)
  If interval(GrMax) >= interval(commonAW) then
     LongestAW = LongestAWInterval(commonAW)
     M[0] = LongestAW.Start
     RefresmentInterval = interval(GrMax)
  Else
     i = 0, j = 0
     While interval(GrMax)*j < Interval(commonAW).end
        If all sources are accessible at interval(GrMax)*j
           M[i] = interval(GrMax)*j
                i++
        j++
Else
  "It is not possible to determine the integration process previously"
```

Figure 4: Accessibility algorithm.

automatically by the system taking care of different criteria (like the maximum level of detail, the no-saturation of the communication resources, etc). As a result, the necessary metadata are obtained so that the DW can be refreshed coherently depending on the type of extraction method and other temporary characteristics of the data sources.

This process does not guarantee that the integration of all of the changes detected in the sources can be carried out satisfactorily. What it guarantees is that the process of integration will be carried out only and exclusively the times that are necessary to obtain the objectives proposed by the DW designer, attending to aspects related to the refreshment and the availability of the data.

4.1 Accessibility Algorithm

Given two data sources, the first task to do is to determine the smallest sequence in the intersection of the set of the availability window values of both data sources that is repeated periodically. We will denominate this concept "*common pattern of availability*". For example, if the availability window of a data source is repeated every thirty six hours and the window of another is repeated every twenty four hour, the "common pattern of availability" will be an interval of duration equal to seventy two hours (see figure 3).

The algorithm, shown in figure 4, first determines the maximum level of detail which both data sources can provide. For example, if a source provides data with a level of detail of a day, whereas

another one provides them at an hour level, it is not possible to integrate them to obtain a level of detail better than a day (hours, minutes, seconds …).

It can occur that the unit (the granule) of the level of detail that can be obtained after the integration of both data sources has a length greater than the "*common pattern of availability*". For example, that a granularity at day level can be obtained and the length of the common pattern is of several hours. In this case, querying the data sources once a day would be enough (it does not make sense to check a data source more often than it is going to be stored). Therefore, the maximum interval width of refreshment in the algorithm is adjusted to the length of the unit of the level of detail, obtained by means of the function "interval" in the algorithm. The value of the period of sampling could be, in the case of the previous example, multiple of a day (two days, three days, one week …). Within the common pattern the moment in which the interval of maximum length begins is chosen to make the refreshment in which both sources are available, so that there is more probability to satisfy the restrictions imposed in the second phase, the *Analysis of Temporal Requirements* (out of scope of this paper). This interval is determined by the "*LongestAWInterval*" function.

In case that the unit (the granule) of the level of detail that can be obtained after the integration of both data sources has a length smaller than the common pattern of availability, it is necessary to determine in what moments within the common pattern both data sources are going to be available to refresh their data. Since it does not make sense to refresh a data more often than is going to be stored, only values that distant the length of the integrated granularity unit are chosen. For example, if the granularity with which the data are going to be integrated correspond to "seconds", the instants will be temporarily distanced one second. Then it is verified that, for all those instants of the common pattern, both data sources are accessible. If it is successful it will be added to the set of instants (M) in which the refreshment can be made.

Some of the instants included in the M set will be discarded in the following phase because they do not fulfil some of the specific requirements that depend on the precise kind of sources. In this case, due to the fact that we are integrating web data sources which usually are simply HTML flat files, we will use a *File Comparison*-based method to do the integration process. This method consists on compare versions of the data in order to detect changes. A more detailed explanation can be found in (Araque *et al.*, 2006b).

Every extracting method has its own requirements. If we are using a *File Comparison*-based method we need to ensure that the following sentence is valid:

$$(ET(DS1) \cup ET(DS2)) \subset (AW(DS1) \cap AW(DS2))$$

where ET(X) is the time needed to extract a change from the source X (*Extraction Time*), AW(X) is the *Availability Window* of the source X and DS1and DS2 are both data sources. In other words, we cannot carry out the integration process of both data sources more often than the time we need to extract the changes. Obviously, if we need thirty seconds to extract the changes from a source and forty seconds to extract them from another source, it is not possible to integrate them every minute because we are not able to get the changes from both data sources so quickly.

Those kind of constraints are checked in this second phase and are better explained in (Araque *et al.*, 2006b).

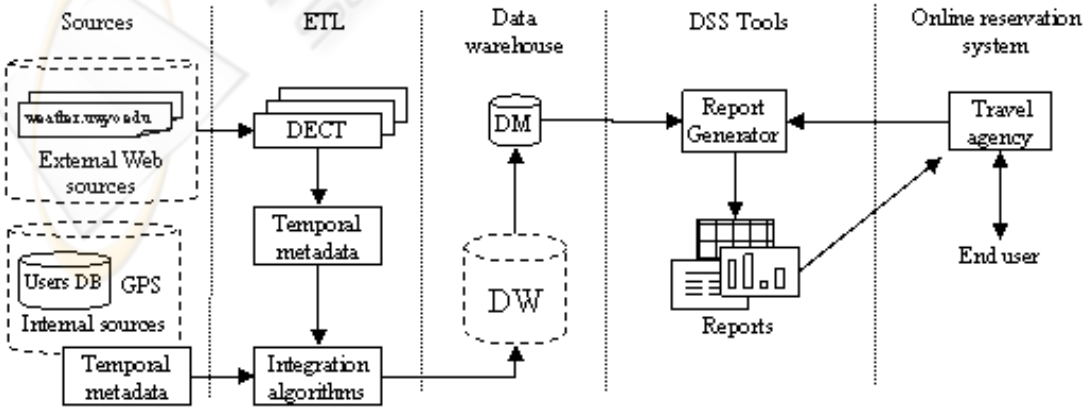

Figure 5: Motivation example applied to tourism area.

5 EXAMPLE

A Decision Support System (DSS) being based on a DW (March *et al.*, 2005) is presented as an example. This can be offered by Small and Medium-Sized Enterprises (SMEs) as a plus for adventure tourism. Here, a DSS (figure 5) is used to assist novel and expert pilots in the decision-making process for a soaring trip (Araque *et al.*, 2006a).

These pilots depend to a large extent on meteorological conditions to carry out their activity and an important part of the system is responsible for handling this information. The DW are filled with a Two web data sources are mainly used to obtain this kind of information:

– The US National Weather Service Website. We can access weather measurements (temperature, pressure, humidity, etc) in every airport in the world.

– In order to obtain a more detailed analysis and to select the best zone to fly, pilots use another tool: the SkewT diagram. The SkewT, or sounding chart, is a vertical snapshot of temperature, dew point and winds above a point on the earth.

The information provided by both data sources is semantically equivalent in certain cases. In order to efficiently integrate these data, it is necessary to use the algorithm described in the previous section. It is needed to use an efficient approach because these kinds of services are offered by SMEs which often have limited resources. The continuous integration of Web data sources may result in a collapse of the resources they use to communicate with their clients, which are not designed to support the laborious task of maintaining a DW up to date.

In our approach, the DW Administrator (DWA) introduces the data sources temporal properties in *DECT* tool (Araque, 2003a, 2003b) and selects the parameters to integrate, for example the temperature. This tool is able to determine the maximum level of detail (granularity) provided by each data source after a period of time. It uses an algorithm to determine the frequency of the changes produced at the data source. We approximate the granularity of the source by selecting the smallest interval that take place between two consecutive changes.

In the first source, the information about the temperature can be precise with a detail of “minute” (for example, that at 14 hours and 27 minutes there were a temperature of 15ºC), whereas in the second case it talks about the temperature with a detail of “hour” (for example, that at 14 hours there were 15ºC). The reason is that in the first source has been detected more than one change within an hour at least once, whereas in the second source all the changes has been detected at least one hour distanced.

It can also determine the time intervals in which this information is available to be queried. Let us suppose that the first data source is always available, but the second one is only accessible from 23:10 to 00:10 and from 12:00 to 15:59 (availability window). Common pattern of availability would include, therefore, a whole day. Applying the accessibility algorithm we would obtain all possible instants of querying in which both sources are accessible and are distanced an interval equal to the maximum integrated granularity unit each other (hourly in the example we are using). Using the values of this example we would obtain {00:00, 12:00, 13:00, 14:00, 15:00}.

For each one of the previous set of instants is necessary to verify that the extraction and integration of the data sources would be possible. For this purpose we use the second algorithm mentioned in the previous section (out of the scope of this paper).

To help DWA in this process we have developed a tool that is able of performing both algorithm described in this paper: *Accessibility Algorithm* and *Analysis of Temporal Requirements*. A capture of this tool can be seen in figure 6.

Using the data extracted from Web data sources a DSS for adventure practice recommendation can be offered as a post-consumption value-added service by travel agencies to their customers. Therefore, once a customer makes an on-line reservation, the travel agency can offer advice about

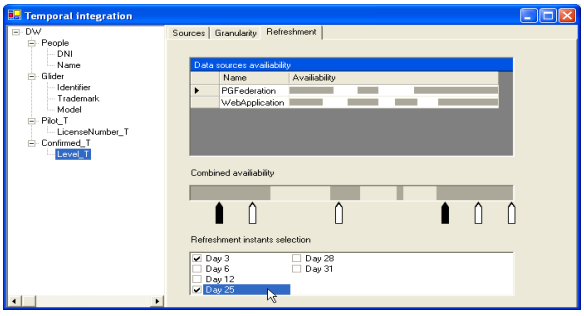

Figure 6: DWA tool for analyzing the refreshment process.

adventure practices available in the area that customer may be interested in. Due to the high risk factor accompanying most adventure sports, a regular information system is far from being accurate. A more sophisticated ICT system is required in order to extract and process quality

information from different sources. In this way, the customer can be provided with true helpful assistance to be aided in the decision-making process.

While logging reservation systems do not need supplementary information as weather forecast, other products in the tourist industry, such as eco-tourism can take a tremendous advantage of last-minute DW. The system constitutes an enhanced modification of the current InfoTours (Araque *et al*., 2006a), an on line availability and reservation system of the travel agency ViveGranada S.L.L. It allows to query a last-minute DW and use the output report to filter the on line availability of outdoor activities offered by the on line reservation system.

6 CONCLUSIONS

In this paper we have presented our work related to an approach for monitoring web sources on the WWW in order to obtain its temporal properties. We use this information for integrating data from different data sources according to the requirements of the DW Administrator.

We also have proposed an algorithm for data integration depending on the temporal characteristics of the data sources. This algorithm determines the smallest sequence in the intersection of the set of the availability window values of both data sources.

In addition, we show an example applied to tourism area where data integrated into DW can be used to schedule personalized travel as a value-added service for electronic commerce.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Research Program under project TIN2005-09098-C05-03 and the by Andalucía Research Program under project 2006/282916.

REFERENCES

Araque, F, Samos. J.: External Schemas in Real-Time Object-Oriented Databases. *20th IEEE Real-Time Systems Symposium*, WIP Proceedings, pp. 105-109. Phoenix. (1999)

Araque, F. & Samos, J.: Data warehouse refreshment maintaining temporal consistency. *5th Intern. Conference on Enterprise Information Systems*, ICEIS´03.Angers. France. (2003)

Araque, F., Salguero, A., Abad, M.M.: Application of data warehouse and Decision Support System in Soaring site recommendation. *Proc. Information and Communication Technologies in Tourism*, ENTER 2006. Springer Verlag, Lausanne, Switzerland.

Araque, F., Salguero, A.G., Delgado, C., Samos, J.: Algorithms for integrating temporal properties of data in DW. *8th Int. Conf. on Enterprise Information Systems* (ICEIS). Paphos, Cyprus. May, (2006)

Araque, F.: Integrating heterogeneous data sources with temporal constraints using wrappers. *The 15th Conference On Advanced Information Systems Engineering*. Caise Forum. Klagenfurt, Austria. (2003)

Araque, F.: Real-time Data Warehousing with Temporal Requirements. *Decision Systems Engineering, DSE'03* (CAISE'03 conference). Klagenfurt/Velden, Austria. (2003)

Bruckner, R., Tjoa, A M.: Capturing Delays and Valid Times in Data Warehouses - Towards Timely Consistent Analyses. *Journal of Intelligent Information Systems (JIIS)*, Vol. 19(2), pp. 169-190, Kluwer Academic Publishers. (2002)

Chaudhuri, S., Dayal, U.: OLAP technology and data warehousing, *ACM SIGMOD Records*, Hewlett-Packard . (1997)

Hammer, J., García-Molina, H., Widom, J., Labio, W., Zhuge, Y.: The Stanford Data Warehousing Project. *IEEE Data Engineering Bulletin*. (1995)

Inmon W.H.: Building the Data Warehouse. John Wiley. (2002)

March, S.T., Hevner, A.R.: Integrated decision support systems: A data warehousing perspective. *Decision Support Systems*. (2005)

Ramamritham, R., Sivasankaran, J. A., Stankovic, D. T., Towsley, M.: Integrating Temporal, Real-Time, and Active Databases, *ACM Sigmod Record*, Vol. 25. No. 1. pp. 8-12. (1996)

Srinivasan, C. L., Ramamritham, R.: Maintaining Temporal Coherency of Virtual Warehouses (abstract). *The 19th IEEE Real-Time Systems Symposium* (RTSS98), Madrid, Spain. (1998)

Watanabe, Y., Kitagawa, H., Ishikawa, Y.: Integration of Multiple Dissemination-Based Information Sources Using Source Data Arrival Properties. *Proc. 2nd Int. Conf. on Web Information Systems Engineering*, Kyoto, Japan (2001)

Yu, L., Huang, W., Wang, S. and Lai, K. K., Web warehouse – a new web information fusion tool for web mining. *Information Fusion*, vol. In Press, Corrected Proof, (2006)