Using intelligent data sources to monitor unusual behaviors in individual's health data

Emma Chávez Mora
Bond University, Universidad Católica de la Ssma. Concepción
emchavez@bond.edu.au

Gavin Finnie
Bond University
gfinnie@bond.edu.au

Abstract

E-health data management is characterized by high pressure and timely access. Accessing patient data requires that all services and objects are connected to make data from different health care sources available. Clinical data warehouses in this context facilitate the analysis, consolidation and access of the data obtained in the patient care process to improve the quality of decision making.

There is a need to capture data events as soon as possible to decrease data analysis latency and maximize the value of information. Given the volume of data that may be generated, some sources can be improved by intelligent local processing and filtering of data for selective reporting. This paper proposes the design of a real time adaptive framework that covers the process of predicting, responding and monitoring unusual behaviors in patient data in a data warehouse environment.

1. Introduction

In health care, clinical data is stored in a variety of heterogeneous and distributed repositories across a range of institutions [1]. Clinical data warehouses (CDW) facilitate the analysis and access to data obtained in the patient care process which improves the quality of decision making and early process intervention [1].

Attributes of a CDW are given by [2][3][4][1] and [5]. CDW are more than a large collection of clinical data and generally data comes from other enterprise systems, devices and sensors with the data being commonly integrated into data marts according to the data warehouse architecture defined. It seems that clinical applications require the construction of a more powerful data model than is usually provided by business approaches. According to [6, p. 8], "in average, patients might have hundreds of different facts describing their current situation". Complicated and unpredictable procedures require quick decisions. Thus, more advanced classification structures are necessary to provide continuous data monitoring and measuring. In summary, it is necessary to manage large amount of heterogeneous electronic heath records, (from general data to RX images) and also give efficient support to process query and analysis at any time (e.g. drug interactions, sensor measurements or laboratory tests).

Electronic patient data is sourced from different organizations such as public hospitals, general practitioners, private clinicians, government and insurance payers [6]. Therefore, a robust architecture that enhances data sharing without duplication is required. It should provide the integration of information from heterogeneous sources, security access to patient's data (security infrastructure) and the access to information in a timely manner (real time). Some of the issues still not fully addressed in the clinical data warehouse domain are timely response by on time data analysis to prevent patient deterioration and sound decision making, security infrastructure and effective data integration.

Clinical Data Warehouses and Data warehouses (DW) in general are refreshed in a periodic manner, usually on a daily basis (during off-peak hours), where the operational sources and the data warehouse experience low load conditions. DWs might impose unacceptable delays due to their batch nature given the cooling-off period between transactions and their representation in the data warehouse, with the most recent data unavailable for analysis as it is caught in the operational sources [7].

The processes to consolidate data from the sources include data export, preparation and transformation, and loading which are usually performed using Extraction, Transformation and Loading (ETL) tools. The value of information for organizations depends on how well timed its delivery is to decision makers. Real time transactions are required to support on-line operational decision making. Organizations aim to use IT solutions to

1530-1605/12 $26.00 © 2012 IEEE
DOI 10.1109/HICSS.2013.589

capture time-critical operational processes [8]. New requirements such as intelligence, quick reaction and adaptiveness are desired for a solution that is capable of consolidating data from a variety of data sources but at the same time allows decision makers to respond to business events at the right time [9].

ETL techniques are difficult to scale up to address the challenge of data loading, performance and low latency for real-time decision support using data warehouses and a new approach is presented in [10]. The framework proposed introduces a new approach for designing real-time DW architectures to support activity monitoring under unusual behaviors in which traditional ETL does not apply. A Transformation, Transference and Loading approach (TTL) is proposed instead. An event driven approach is used as a way to sense and react in real time to changes in environment conditions. Data is pre-filtered, processed and analyzed at the sources by enabling learning capabilities in them. Thus, sources of information sense and react by "pushing", rather than "pulling", data into the warehouse only if needed.

The design proposed is being tested in e-health by developing an architecture that can actively monitor coronary heart disease patients and react when abnormal patterns occur in the patient data. Coronary heart disease monitoring of patients takes place at each step of patient management, from disease detection to disease prognosis and from surgery to recovery. During routine screening, day to day measurements (sensor devices) and patient knowledge can be obtained and risk scenarios can be detected. Health information is widely distributed and involves many different processes and interactions. It contains individual patient information collected and stored by a person (a health professional for example) or a combination of many sets of individual information collected by more than one entity (hospitals, private institutions, health funds and others) which is stored in different locations in large databases.

During monitoring real time response is kept to minimum latency by eliminating the data availability gap to perform data analysis which enables health care organizations to concentrate on accessing and processing valuable/meaningful patient data only.

The underlying rationale for this research is that the TTL approach provides a more effective architecture for monitoring health data than existing ETL architectures. The following section provides some background on prior research on data transformation and loading in data warehouses. Section 3 explains the TTL approach and its application in health care, the concepts of pre-analysis and transfer and the agent based architecture. Section 4 describes the validation of the approach from both effectiveness and efficiency while section 5 provides some discussion of the research outcomes.

2. Background

The nature of health data makes its management complex. At high levels health information is used for policy, strategic planning, research and education. However, at low levels, it can be used for individual health services such as treatments and prescriptions. A major problem in this area is data management in the sense of being able to access and use heterogeneous information (different data structures and data types for example) to provide on time care and rapid response to clinical decisions. Clinical data warehouses (CDW) in this context can facilitate the analysis and access to data obtained in the patient care process to improve the quality of decision making and to help with patient data consolidation by assisting with the extraction of only that data which is valuable for patient monitoring [11].

Nowadays pervasive health care monitoring environments, in the same way as in business, gather information from a variety of data sources, but they include new challenges because of the use of body and wireless sensors which makes systems more complex to monitor in real time. Business Intelligence (BI) architectures refer to technologies, tools and practices to integrate, process, and present large volumes of information. Although linked to business, the tools and techniques are applicable to any large scale data analysis situation. The use of BI tools are still very limited in healthcare given the generation of false positive alerts which holds up patient specific data processing in right time for clinical decision making [12][13].

Traditional clinical data warehousing architectures utilize one or more on-line transaction processing (OLTP) databases before data is moved into a data warehouse on a monthly, weekly, or daily basis. Usually the patient data is processed in a staging file before being added to the data warehouse. In general patient data must be loaded regularly in order to facilitate the monitoring process. To perform this operation, data from various operational systems needs to be extracted and copied into the warehouse before reaching the reporting stage. Data needs to pass through at least four main stages: data extraction, data transformation which includes data verification, data cleaning, data integration and aggregation, before consolidation in a warehouse for further analysis and data reporting.

BI applications put together processes related to data integration, data storage, and knowledge

management with analytical tools with the aim to enhance the quality of inputs that ideally should arrive at the right time, right place and in the right form to advise decision makers [14].

Most BI architectures monitor crucial business operations by using a data warehouse. Raw data from several complex sources can be consolidated to provide high quality and consistent information [15][16][17].

In data warehouse environments operational processes move data from operational sources to the warehouse. This includes data export, data preparation, and data loading usually performed by Extraction, Transformation and Loading (ETL) tools. ETL functions are a critical component of a BI solution as they are responsible for retrieving data from different sources, preparing it by normalizing and transforming it in some way to be inserted into some other repository (usually a data warehouse) to be ready for analysis and reporting [18][19].

Traditional data warehouses systems may impose unacceptable delays as they are out-of-sync mainly because of their batch nature. ETL techniques have not been scaled up to address the challenge of data loading, performance and low latency to provide real time decision support systems.

The use of intelligent techniques for data preparation and loading may considerably improve the overall process of data warehouse population. The design of the back end of intelligent ETL processes is becoming important but at the same time complex, as problems of optimization and modelling emerges. During the last decade a small number of research efforts were carried out to monitor data in real time and to address the challenge of ETL process optimization. Some of them can be seen in research conducted by [11][20] [21] whose approach tried to break the gap between the time at which operational data is created and the time the analysis information becomes available. The need for intelligent data management is relevant to a broad range of data warehouse applications, including healthcare and business activity monitoring.

In research to achieve real time data warehousing [22] presents a technique to distribute the Online Transaction Processing (OLTP) source data that has to be extracted. The ETL refreshment is done off-line for some tables and for others in a near real time manner to create a near real time data warehouse methodology. In architectures like SARESA [21] the authors argue that they have built a real-time warehouse architecture. The data changes are captured by means of data capture web services and the real time data is achieved by using a multi level data cache.

After reviewing research conducted in real time and active data warehousing technologies for BI, it appears that only near right time has been achieved and most of the techniques studied focus on the "time" response from the "machine" point of view to deliver the information rather than on improving the speed to analyse data and therefore, provide advice at the right time to decision makers.

Data analysis is performed only once data has been consolidated; therefore reporting and alerts are the last activities after some sort of ETL has been done. Moreover, real time data warehouse approaches usually query for data and capture a vast amount, so data is pulled (extracted) from the sources of information to then be processed in a consolidated repository. Furthermore, nowadays applications are generally still human administered with no possibility of self-adaptation. The idea of dynamic adaptation to healthcare or business requirements (a logical adaptation rather than a structural adaptation) of a DW architecture has not yet been proposed.

Consequently, the demand for BI solutions and frameworks is continuously growing but information systems research in the field "is to be charitable, spare" [14]. Most of the techniques in use have to schedule the data extraction and all of them analyse the data only when it has been consolidated in the data warehouse. DWs have done a good job to pull together large volumes of historical data but they are usually not built to deliver real time information [23].

There are no ETL tools that can provide real time data processing efficiently. There has been no mechanism proposed that can continuously feed and update data warehouses as soon as data changes in the data sources [15] [23] [24]. Traditional business intelligence and data warehousing technologies "do not directly address time sensitive monitoring and analytical requirements" [23]. Typically, a DW is designed and built from historical information with the data being refreshed daily at best. It is argued that the process to access up to date information requires that all the services and objects are connected to make data from the different data sources available with a minimum latency possible. Being able to respond quickly to business events denotes the difference between success and failure in areas such as health care, stock markets, food manufacture, logistics and others.

From the healthcare (and business) side there is a need to capture data events as soon as possible to maximize the value of information [9] so that a quick response from the monitoring systems can help decision makers to take sound decisions that can make a difference to respond in the right time e.g. to

save lives because a risk is present or in business to detect frauds or move products to quick sale.

From the research side, there is as yet no smart framework with some sort of adaptation to environment requirements capable of dealing with real time data integration from a variety of data sources and with real time alerting. For this reason, *the design of a framework that covers the process of monitoring relevant individuals data and responding to unusual behaviours in real time to decrease the time of data analysis, and enhancing the value of information while using data warehouses is proposed.*

3. The Transference, Transformation and Loading approach –TTL

The TTL architecture works with the idea of monitoring meaningful data to identify and react to unusual scenarios only. Thus, the monitoring mechanism is based on alerting and reacting when abnormal behaviors are present in the data. The architecture monitors a subset of the data that is captured in the data sources.

That subset, any relevant entity to monitor at the source level, is called a compound for this work. The structure of a compound is based on the set of inter-related components considered important for the specific problem being monitored and is defined by the application. Sectors are the specific application domain i.e. for this case monitoring of coronary heart disease patients. That compound will be monitored and pre-analyzed based on a pre-defined base of knowledge that gives the boundaries for the monitoring mechanism. In this way entities that perform differently but belong to the same sector will be monitored. An example of a compound and sector can be seen in Figure 1.

The knowledge to monitor comes from the area of interest to monitor. In this case general knowledge comes from the sector which is coronary heart disease patients. The general knowledge has the rules and the standards that apply to the ongoing assessment of coronary heart disease patients, and that affect the way a compound should behave. Compound knowledge is formed from all the accepted ranges, episodes and/or critical data to monitor for a particular patient. That information comes from the knowledge obtained in all the historical data available for each entity that resides at the sources of information.

In our case general knowledge about which main features (symptoms and combination of symptoms) to consider in heart disease was taken from list of 24 clinical features obtained by research conducted by [25]. That list provided us with the main critical diagnostic features for five major heart diseases which could be considered the most relevant for diagnosis and ongoing assessment. A general expert panel of four cardiologists reduced this to a sub list of 13 symptoms considered important to monitor in ongoing patient assessment for patients identified with coronary heart disease as well as defining their relative level of importance.

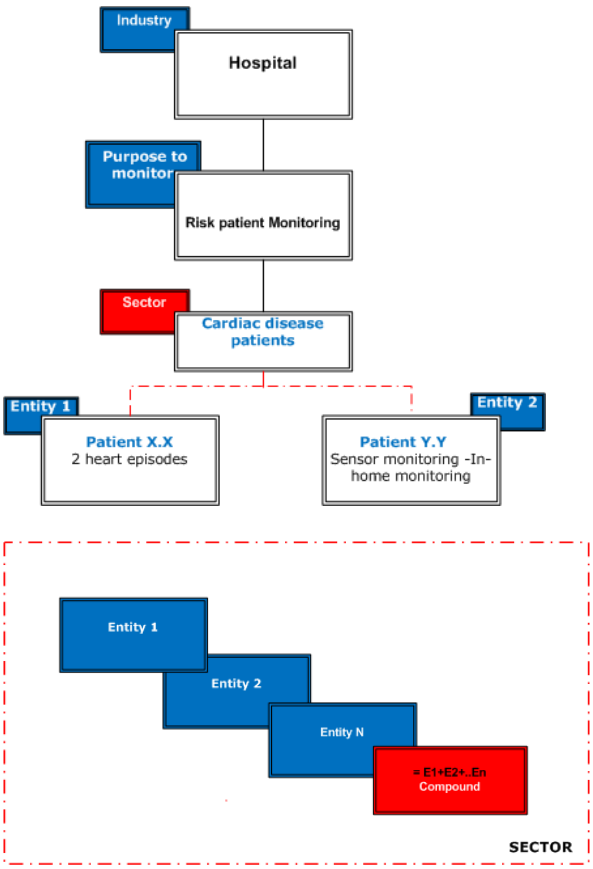

Figure 1. Sector, entities and a compound.

Table 1 and 2 give an overview of the sector general knowledge.

Rather than providing a traditional BI architecture in which a centralized mechanism is implemented to start with the data analysis necessary, the TTL architecture splits the data analysis in two main areas and distributes the decision making in two units. The Local decision making unit enables data pre-filtering and alerting as soon as relevant data to monitor has been detected from the sources of information and pushes data to the warehouse for further storage and/or decision making. The Centralized knowledge management unit on the other side drives the updating mechanism for the knowledge that the local

decision making area uses and helps to go further with the data analysis whenever it is needed (See Figure 2).

Table 1. General knowledge (symptoms and level of importance)

Level of Importance	Symptoms
High importance	ST-T alteration Dyspnea Hypertension Discomfort, heaviness in the chest Chest Pain Neck venous return or engorgement
Medium Importance	Cyanosis Systolic murmur Dizziness Diastolic murmur
Low importance	Headache Second heart sound Upper respiratory infection

Table 2. General knowledge (Episodes)

Episode	Symptoms
Pathological cardiac remodeling	Cardiac enlargement + Dyspnea + Systolic murmur
Cardiac insufficiency	ST-T alteration +Chest pain
Malignant ventricular arrhythmia	Syncope + Palpitations
Myocardial infarction	ST-T alteration + Third heart sound
Ischemia of the papillary muscle	New systolic murmur + ST-T alteration
Right ventricular infarction	Cervical venous engorgement + ST-T alteration + Chest pain

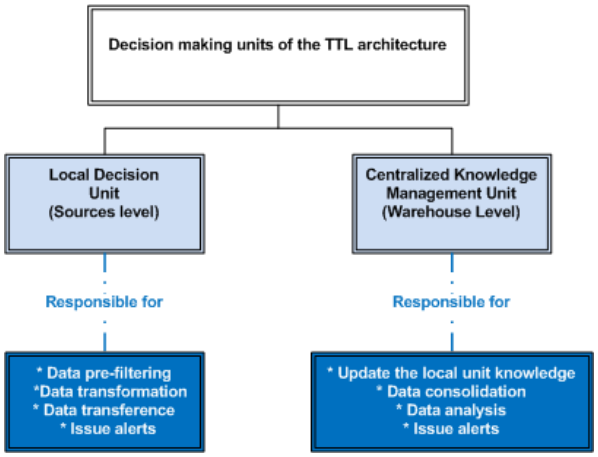

Figure 2. TTL main functions

As seen in Figure 2 there are five main functions performed in the TTL architecture; pre-analysis/analysis, transformation, transference, alerting and data updates. These have been organized in layers/tiers to perform the monitoring process and to start pre-analyzing patient data from the very beginning. To deal with latency issues the TTL architecture empowers the data sources with intelligent capabilities, using compounds and general knowledge, to monitor and pre-analyze critical data. The pre-analysis is performed by using a Multi Agent System Architecture (MAS) which for the prototype was implemented with C code and classes. Agents in the architecture learn and reveal patients data activity patterns through day to day measurements and the data history contained in each source of information

3.1 Pre-analysis and Transformation

To provide integrated access to a variety of heterogeneous sources such as data bases and sensor devices, the pre-analysis and transference module is responsible for monitoring entities (Patient1...PatientN) in each source. We argue that there is enough knowledge already in operational systems to extract and use to empower the sources to monitor meaningful data changes. By achieving a certain level of autonomy and intelligence employing agents, a push system with local empowerment, rather than a fully centralized approach, can be deployed. Here the mechanisms for pre-analyzing and pushing data to the central repository are established. Therefore, this module monitors meaningful data contained in each source and reacts to abnormal ranges for a patient. To do that;

- Source agent (SA) is subscribed to the patient ID of the data to monitor in each source of information.
- Then, through a set of specific rules that SA has from the base of knowledge (historical data) of the sector, it compares normal parameters to take the decision to perform an action such as deliver information to the learning repository, to send an alert to data managers, or not do anything as the data captured even though is relevant to monitor is within normal parameter ranges.

Source Agent does not monitor all data available in each source. It monitors the changes in the accepted ranges and episodes of just particular patients in sources such as the hospital DB, the GP database and the data that is coming from the sensor devices that the patient is using at home.

As soon as valuable data changes are captured at any source information a trigger alerts to SA which checks:

- IF the data change (dc) in the source is meaningful data to monitor THEN check it against datastructure.type accepted for the compound
  - IF dc.type!= datastructure.type accepted THEN react/alert ENDIF
  - IF dc.type = datastructure.type.accepted THEN check dc against accepted ranges
    - IF dc = accepted.range.feature THEN go to sleep because no action is needed ENDIF
    - ELSE react/transform/transference ENDIF

This pre-analysis provides the opportunity to take decisions and to alert as soon as abnormal behaviors have been detected and before all relevant data is consolidated in the central repository.

Transformation is one of the main stages and time consuming activities while integrating data from heterogeneous data sources. It is needed as data structure standardization across different organizations and data sources might be impossible to achieve. Different data sources could have different keys for representing the same entity, entities could contain unnecessary attributes and/or fields which are irrelevant to monitor. Data therefore needs to be transformed before being transferred.

In our approach data might be transformed when the dc's type is accepted but the content of dc is not in normal ranges. The transformation technique to be used, such as conversion, filtering, sorting and translating, will always depend on the data to be integrated from the sources. Some sources might need more or less transformation work than others.

Data is transferred to the central repository when there is not enough knowledge to take an action at the source level and a consolidated view of data between the general knowledge and/or other entities data is needed. A view of these will be explained in the following section.

3.2 Data transfer/update

To achieve a level of data warehouse adaptation Source Agents can learn from the changes made at the source level. Thus, repetitive changes in behaviors might mean an update of the knowledge or a change of requirements for the patient to monitor. These will allow determining and reacting to new requirements, changes in the disease parameters, and knowledge compound changes, such as range of blood pressure or weight, providing logical adaptability for the architecture.

3.3 Data alerts

Once data has been analyzed, whether this analysis is performed in the central repository or at the source level (Figure 3), alerts are sent to the decision makers because abnormal behavior in the content data of a particular patient has been detected.

Checking unusual behaviors (dc) allows the Alert Agent to perform one of the following:

- IF the dc indicates high or medium importance the decision maker is notified of the level and the dc saved
- IF the dc indicates a data error or unknown behavior the user source is notified.

A classification of changes by level of importance was needed as a way to obtain a risk factor for each patient to know who needs the alert and when it needs to be issued. As an example alerts could be sent to a patient to warn that the sensor device that it uses may not be working properly as the data pre-analyzed by the source agent seems to be incomplete or not in normal ranges.

Moreover, alerts could be also sent to the decision makers given the behaviors represent a risk for the entity to monitor. Furthermore, if the data in the consolidated repository represents a new episode for the patient the alert is sent to the health care practitioner responsible for the patient.

4. TTL validation

4.1 The Local Decision Unit Prototype

To validate the TTL approach a simulated environment of coronary heart disease patient scenarios (compounds knowledge), disease data and cases was created and deployed. A PostgreSQL data base was used which contains simulated patient's data, episodes and cases from the last 10 years for 20 coronary heart disease patients. A PostgreSQL data warehouse was also built which contains a multidimensional view of patients that have presented episodes, unusual and relevant for disease prognosis events, in a determined time.

The local decision unit was implemented as a prototype using C. The main characteristic of the local decision unit was to be able to capture the changes made (update and/or insertions for example) of all the relevant data defined to be monitored at the

source level. Once relevant data changes have been detected and captured a pre-analysis is required. The pre-analysis filters the data that the warehouse will contain and according to pre-defined decision rules is able to recommend as soon as knowledge becomes available what might be the best decision to take according to the findings.

For the implementation of the local decision unit a PostgreSQL relational data base was built for maintaining patient data. PostgreSQL was chosen primarily because is a non-commercial application, multi-platform, and it does provide enough documentation and capabilities as a DBMS to build a prototype for research purposes.

The implementation focused in creating a mechanism that allowed source agent to capture changes relevant to coronary patients and pre-analyse them. The function *pg notify* (text, text) enabled a simple pushing mechanism to notify data changes by providing a publish and subscribe mechanism within PostgreSQL. The data source was empowered to send patient data changes to source agent. A source agent class was developed using C to listen to patient notifications. As soon as a data change is received, the agent checks whether the change was relevant for the patient being monitored. If yes, it proceeds with the predefined rules to pre-analyse the change and decides what action (if any) is needed e.g. providing an alert by sending a message to the health care staff responsible for the patient (decision maker) and/or inserting the data in the data warehouse.

In a system designed to actively monitor data such as healthcare, resource consumption and response time are usually the metrics of interest to evaluate a system. These can be evaluated from two key aspects, which are efficiency (the way to do the task using less resources) and effectiveness (doing the task in the right way). The main results of the local unit implementation in terms of efficiency and effectiveness are provided in the following sections.

4. 2 Computational efficiency

A dedicated machine (Intel core Duo, 3GHz and 3.25GB of RAM) was used. A PostgreSQL data base with 20 heart disease patients, with records from the years 2000 to 2010 was simulated. Data in the database includes GP patient records (visits, symptoms, outcomes/events, and medication), pathology results (test, ECG results), hospital patient records and sensor monitoring data results). That gave us an environment of around 6000 records for the total of patients.

We used the Linux Ksar utility to monitor the system states and resource usage during the tests in order to compare traditional data extraction vs intelligent transference and pre-analysis. The results of these tests can be seen in Table.3.

Table 3. TTL computational efficiency

	Scenario	Duration in Sec.	Average usage of CPU(%)		
			User	System	Idle
Traditional extraction	Simple	1.027	50.29	19.17	30.44
	Complex	1.005	51.27	18.29	24.52
Intelligent Transference	Simple	0.637	30.27	21.15	48.58
	Complex	1.297	44.17	19.31	36.52

Traditional extraction was considered as a simple planned and batched programmed query that extracts patient's data to be moved to a warehouse in a certain period of time. Intelligent transference is the new approach that we propose. A simple scenario refers to a scenario in which data has changed at the source level but is not relevant to monitor. A complex scenario refers to a new data entry at the source level that is relevant to monitor and an action must be taken.

As can be seen in the table the intelligent transference, (agent pre-analysis) consumes less than or almost the same amount of CPU as the traditional data extraction mechanism. Therefore, enabling pre-processing and filtering at the source levels does not stress the sources of information. Source agent seems to run to a low priority (given the % CPU idle used) so does not impact programs that run at normal priority in the server.

In the complex scenario SA performed pre-analysis and transference and it did not show mayor differences in relation to traditional transference.

These tests were planned to move a number of KB only (200 records) which is the best scenario and it does not represent a Very Large Data Base (VLDB) environment. Nevertheless, in a VLDB environment traditional data extraction will use more machine resources and it will be more time consuming [20] [26]. In our case the database size is not relevant as SA will keep almost the same average consumption as it will only analyze relevant data only (a small data load) and not all the data that has changed in the source. This approach can thus handle large numbers of patients without system degradation.

In terms of process duration traditional data extraction is affected by the number of rows/bytes to extract while intelligent data transference is affected

by the amount of data to analyze. Because in the tests a small data load was simulated, it was not really possible to compare process duration with the two scenarios proposed. Although this data is included in the table they were not considered enough to be discussed in any depth here and further tests with a large dataset need to be done in the future.

4. 3 Decision making Efficiency

A panel of 8 cardiologists was used in testing and validating the efficiency of the TTL framework prototype to support decision making in health care. We provided them with a test worksheet to validate 4 main areas of coronary heart disease and a view of how the prototype worked to monitor a patient in a test of 4 different scenarios.

The test scenarios were for providing alerts for:

- High behavior change
- Medium behavior change
- Data error
- A patient episode

The 4 main areas to validate were:

- improving quality of care
- improving quality of life for patient
- saving time of healthcare professionals
- modernizing health care delivery

These areas were taken from [27] and [28] who provide an overview of the areas to consider for validating the effectiveness, quality and other ways of improving performance of health systems in e-health.

Improving quality of care (IQC) was defined by whether the TTL prototype provides easier and faster access and the right patient data. Improving quality of patient life (IQP) was defined as to how well the TTL prototype monitors patient's heart disease conditions and consolidates individual health records to deliver tailored treatments. Saving time (ST) was defined as the TTL prototype providing a supporting decision making mechanism to health care staff. Modernizing health care delivery (MHC) was defined as TTL bringing a degree of sophistication to the health care systems by allowing a faster flow of information for continuous care.

The main results are displayed in Table 4. Table 4 shows the percentage acceptance rate of the TTL prototype among the expert panel for all the test cases considered. TTL performs well in terms of efficiency of decision making with the main success being to identify errors and patient episodes.

Table 4. TTL decision making efficiency - Acceptance rate

Scenario	IQC (%)	IQP(%)	ST(%)	MHC (%)
Blood pressure ↑	50%	62.5%	37.5%	75%
Dizziness=yes	50%	100%	37.5%	75%
Data error	87.5%	100%	100%	100%
Blood pressure + Dizziness	100%	100%	100%	100%

According to the expert panel the identification of individual abnormal behaviors for tests 1 and 2 (High and medium importance change of behaviors) needed more refinement as the action to take for a change of symptoms depends on an individual's environment conditions given the outcome could represent a false positive alert. Hence, a patient blood pressure could rise because of movement, stress and other factors which do not really represent an emergency. The panel could not completely agree on the efficiency of the alert mechanism tool to detect these types of scenarios and only the detection was accepted as efficient.

Therefore, the inclusion of environmental conditions that might affect an unusual behavior might need to be included in the future as part of the whole framework in order to not give false positive alerts.

4.4 Effectiveness

We argue that by monitoring key features in each patient, pre-analyzing these and then if needed transferring them to a central repository the proposed architecture is more effective than traditional ETL and data warehouse architectures.

By having local knowledge and empowering the data sources (using a monitoring agent) we have reduced the number of steps required to analyze data. Data analysis has been moved to an early stage starting once relevant data to monitor has been changed in a source. That pre-analysis is performed now in the local source. If there is enough knowledge to perform an action, an alert mechanism is activated to either alert the patient because maybe the data that has changed in the source is incomplete (e.g. sensor device data) or alert the health care staff because the data that has changed is relevant to monitor and implies that the patient might be at risk.

Data inconsistency is detected in an early stage because of the knowledge about data types and data structure that Source Agent has in each source of information. Therefore, SA knows data structures and

types of each source. Thus, data inconsistency is picked up in relevant data as soon as the agent is notified. Source Agent checks data structures and value before performing any analysis which helps us to inform and not include uncompleted data as part of the analysis.

By monitoring only relevant data in a distributed environment and by having local knowledge to check normal patterns it is possible to identify changes in the disease prognosis at an early stage and send an alert as soon as possible to the health care staff in patient risk scenarios.

5. Discussion and conclusions

We have proposed a new approach to designing clinical data warehouses to manage clinical data to a patient level in which traditional extraction, transformation and loading tools do not apply. As the data warehouse is not programmed for querying the sources for the information, the transformation, filtering and information cleaning will be performed by agents in each data source.

As discussed in sections 1 and 2, most monitoring architectures that use a data warehouse analyze data only once it has all been consolidated in the central repository. We argue that there is enough knowledge already in operational systems to take decisions at a local level. By using an agent infrastructure as discussed in section 3, agents learn and respond to patient events to be able to react to abnormal behaviors as soon as data is captured at the source level. In this way, important time for decision making is saved. The decision making starts from the owner of information and only useful data (data that needs to be monitored) is sent to a consolidated repository. Not all the agent framework has been discussed here with only those agents that are involved in data management being mentioned. However the prototype outlined in section 3 provided a test-bed to evaluate the efficiency and effectiveness of the approach. Section 4 discussed these measures and showed that the TTL approach offers a better way of providing near real time response to monitoring critical conditions in coronary patients as well as improving decision making efficiency.

In a typical operational healthcare system there would be thousands of patients with diverse sources of information. In a consolidated system, data extraction will be performed for all these thousands of records before proceeding with data analysis. If there is not enough information to proceed with the analysis, it will only be detected once data has been collected and sent to the central repository. There is a significant amount of data that needs to be extracted before it can be analyzed.

The TTL architecture is:

- *Intelligent,* to detect abnormalities and react as soon as relevant changes have been captured at the sources of information.
- *Active*, to continuously capture meaningful events, and to inform decision makers as soon as possible.
- *Agile*, to react and provide alerts in the shortest time possible.
- *Adaptive*, in the sense of providing an infrastructure that learns from abnormal behaviors and historical operational data in order to propose changes and updates in the knowledge base and logical structure of the rules.

By monitoring key behaviors, evaluating and then responding only to abnormalities the architecture should be able to respond and/or alert to risk scenarios in a more effective way than traditional right time data warehousing strategies.

6. References

[1] T.R. Sahama and P. R. Croll. A datawarehouse architecture for clinical data warehousing. In Proceedings of the fifth Australasian symposium on ACSW frontiers, Australia, 2007, pp. 227 – 232.

[2] E. F. Ewen, C.E. Medsker and L.E. Dusterhoft. Data warehousing in an integrated health system; building the business case. In 1st ACM international workshop on Data warehousing and OLAP, United States, 1999, pp.47 – 53.

[3] E. M. Kerkri, C. Quantin, F.A. Allaert, Y.Cottin, P. Charve, F. Jouanot, and K. Yetongnon. An approach for integrating heterogeneous information sources in a medical data warehouse. Journal of Medical Systems, 25(3), 2001.

[4] C. S. Ledbetter and M.W. Morgan. Toward best practice: Leveraging the electronic patient record as a clinical data warehouse. Journal of Healthcare Information Management, 15(2), 2001, pp.119 – 131.

[5] N.B. Szirbik, T. Pelletier and C. Chaussalet. Six methodological steps to build medical data warehouses for research. International Journal of Medical Informatics, 75(9), 2006, pp.683–691.

[6] P. Chountas and V. Kodogiannis. Development of a clinical data warehouse. Proceedings of the IDEAS Workshop on Medical Information Systems, 2004, pp. 8–14.

[7] M. J. Huang, M. Y. Chen, and S. C. Lee. Integrating data mining with case-based reasoning for chronic diseases prognosis and diagnosis, Expert systems with applications, 2007, pp. 856–867.

[8] T. Jaorg and S. Dessloch. Near real-time data warehousing using state-of-the-art ETL tools. The 3rd International Workshop on Enabling Real-Time for Business Intelligence, 2010, pp.1-16.

[9] R. Hackathorn. The BI watch: Real-time to real-value. DM Review, 2004, pp. 1-4.

[10] E. Chavez-Mora and G. Finnie. TTL: A Transformation, Transference and Loading Approach for Active Monitoring, Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science, 201, pp. 124-135.

[11] A. D. Kemal, D. Laurent, and D. Umeshwar. Towards an architecture for real time decision support systems: challenges and solutions. Proceedings of the international database engineering and applications symposium, 2001, pp. 303 – 311.

[12] S.R. Chowdhury, A. Birswas and R. Chowdhury. Design, simulation and testing of an optimized fuzzy neuronal network for early critically diagnosis. In IEEE first international conference on emerging trends in engineering and technology, 2008, pp. 665-670.

[13] S. Ferdous, L. Fegaras, and F. Makedon. Applying data warehousing technique in pervasive assistive environment. Proceedings of the 3rd International conference on pervasive technologies related to assistive environments, Greece, 2010.

[14] S. Negash. Business intelligence. Communications of the Association for Information Systems, (13), 2004, pp.177-195.

[15] U. Dayal, M. Castellanos, A. Simitsis, and K. Wilkinson. Data integration flows for business intelligence. Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, 2009, pp.1–11.

[16] J. Kang and H. Han. A business activity monitoring system supporting real-time business performance management. IEEE third International Conference on Convergence and Hybrid Information Technology, 2008, pp. 473–478.

[17] B. Azvine, Z. Cui, D. Nauck, and B. Majeed. Real time business intelligence for the adaptive enterprise. Proceedings of the 8th IEEE International Conference on E-Commerce Technology, 2006, pp. 29–40.

[18] A. Simitsis, K. Wilkinson, M. Castellanos, and U. Dayal. Qox-driven ETL design: reducing the cost of ETL consulting engagements. Proceedings of the 35th SIGMOD international conference on Management of data, 2009.

[19] M. Golfarelli and L. Rizzi, S. And C. Cella. Beyond data warehousing: What is next in business Intelligence? In Proceedings of the 7th ACM international workshop on Data warehousing and OLAP, 2004, pp. 1–6.

[20] Y. Ying and L. Wen-Sian. Correlation aware synchronization for near real time decision support systems. Proceedings of the 13th International Conference on Extending Database Technology, 2010, pp. 39–50.

[21] T. M. Nguyen, J. Schiefer, and A. M. Tjoa. Sense & response service architecture (SARESA): an approach towards a real-time business intelligence solution and its use for a fraud detection application. In Proceedings of the 8th ACM international workshop on Data warehousing and OLAP, 2005, pp. 77 – 86.

[22] M. Javed and A. Nawaz. Data load distribution by semi real time data warehouse. Proceedings of the 2010 Second International Conference on Computer and Network Technology, 2010, pp.556–560.

[23] W. Eckerson. A business approach to right-time decision making, 2006.

[24] T. Chieu and L. Zneg. Real time performance monitoring for an enterprise information management system. IEEE international conference on e-business engineering, 2008, pp. 429–434.

[25] H. Yang, J. Zheng, Y. Jiang, C. Peng and S. Xiao. Selecting critical clinical features for heart diseases diagnosis with real-coded genetic algorithm. Applied Soft Computing, 2008, pp. 1105–1111.

[26] M. Nascimento, T. Zsu D. Kossmann, R. Miller, J. Blakeley and K. Schiefer (eds.). Proceedings of the 30th International Conference on Very Large Databases. Morgan Kaufmann, San Francisco, 2004.

[27] Council of the European Union. Council Conclusions on Safe and efficient healthcare through eHealth, 2980th employment, social policy, health and consumer affairs Council meeting, 2009.

[28] O. Arah., N. Klazinga, D. Delnoij, A. Asbroek, and T. Custers. Conceptual framework for health systems performance: A quest for effectiveness, quality, and improvement. International journal for quality in health care, 2003, pp. 377-398.