HOW TO BUILD A MODERN PATIENT CARE APPLICATION

Dieter Gawlick, Adel Ghoneimy and Zhen Hua Liu
Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065, U.S.A.

Keywords: Declarative programming, Data management, Knowledge management, Tagging, Meta-queries.

Abstract: Patient care is typically supported by several incompatible applications in terms of their data models and semantics, ranging from support for surgery, to ICUs, to standard health care. These programs are focused on specific situation and give doctors a very limited view at patient information. This paper argues that the IT technology has evolved to a point that it is now possible to develop a generic patient care application that manages all patient data for all situations. Furthermore, this application provides a framework for capturing medical knowledge. With this knowledge the application is able to extract medically relevant information from data, even if the extraction is outside of the medical expertise of a doctor or if the extraction is outside of the capability of the human brain. Medical knowledge can be customized by domain, personal views, and patient condition. Additional benefits are: alerting the right doctor of time critical situations, identifying relevant information from a (new) patient's history, and timely sharing of (new and codified) medical knowledge and experience across the medical community. The ideas are based on prototype work between the University of Utah, Health Science Center (UUHSC), the University of Coimbra, and Oracle (Guerra et al., 2009).

1 INTRODUCTION

Today's patient care applications are focused on managing patient data using only the most basic functionality of database systems. Operational tasks, such as monitoring patient conditions, alerting doctors for critical patient conditions, classifying and extracting information from patient data and developing and supervising treatment plans, are considered to be beyond the scope of DBMS. Therefore, a significant amount of procedural code is written leading to limited functionality, lack of customization, and extremely high development and maintenance costs. Long development cycles are another side effect.

In this paper we show that modern database technology has matured to a point that even operational tasks can be handled by databases based on declarative specifications.

Databases can capture a wide variety of data since they support variety of data models; SQL for the management of highly structured data, XML for the management of semi-structured data (Liu et al., 2005), and RDF/OWL for formal semantics and reasoning (Das, S. et al., 1004); DICOM can be used to manage medical images. Ad-hoc access to single records is complemented by a rich support for analytics, all the way to the support for the development of (non-hypothesis) driven predictive models and machine learning. Databases are also able to automatically manage the history of patients providing easy and fast access while minimizing storage consumption; updated and deleted records are simply remembered.

Databases have a variety of ways to extract information from raw data in near real time. Examples are triggers, real-time log-miners, registered queries, online scoring, and RDF/OWL rules (Oracle, 2010b). Any of these technologies can be seen as a way to capture domain knowledge and apply them to the data; i.e., databases not only manage data but also manage knowledge and apply this knowledge immediately to incoming data (Motro and Yuan, 1999). Extracted information can be stored in the database as well, however, extracted information will typically be represented by an abstract vocabulary that most likely consists of a set of tags; e.g., the vitals of a patient are critical. The association of tags to data is often the core of applying domain knowledge to captured data. Time critical information will trigger alerts. New or improved knowledge will trigger a re-evaluation of existing data and trigger alerts as well.

Gawlick D., Ghoneimy A. and Liu Z..
HOW TO BUILD A MODERN PATIENT CARE APPLICATION.
DOI: 10.5220/0003298404270432
In *Proceedings of the International Conference on Health Informatics* (HEALTHINF-2011), pages 427-432
ISBN: 978-989-8425-34-8

Knowledge is widely shared among experts; however, it is important that knowledge can be tailored to the view of sub-domain specialities and individuals. This can be done by integration of knowledge management and consumer management.

The rest of the paper is organized as followings. Section 2 describes the motivating example of patient care. Section 3 describes important aspects of advanced data management, knowledge management, and consumer management, section 4 shows how these technologies are used synergistically, section 5 compare this approach to others and section 6 draws conclusions.

2 PATIENT CARE

The capturing of patient information with EMRs is on its way to become the baseline for using IT technology for patient care. Ideally, the medical personnel and patients should have unobstructed access to any information they are entitled to see; e.g., they should have access to individual records – present and past as well as support for higher level functions such as trends and scoring against any number of conditions.

This management of and access to patient records has to be complemented by the management of the medical knowledge, i.e.., the system has to support the medical personal and patients in the interpretation of data. Here are two examples: the application should be able to determine if a specific value, or a set of values indicate a risk, i.e., create if necessary a tag indicating a serious or critical condition. The application should also to able to identify those models – diagnosis's - for which a patient scores high. This application of the medical knowledge has to be done whenever new data becomes available. If there is any urgency in the computed information, the right members of the medical personnel have to be alerted.

The interpretation of data is obviously a very complex process; it depends not only on patient data but also on the diagnosis, the speciality, as well as the preferences of a doctor. Consequently, the knowledge management has to capture shared and generally accepted knowledge and also provide the freedom for individual points of view.

The knowledge that is used to support an individual doctor can be considered as a profile of that doctor. Profiles should be used to significantly improve query processing. Instead of looking just at data, doctors should be able ask for important information that are (hidden) in the patient data; as of the time of the query or in the past. This would allow doctors to find important information fast, even in a large amount of data. Missing data (lab-test) should also be identified.

Since the medical knowledge evolves extremely fast; the application must not only be able to adapt easily to new and improved knowledge but also help doctors to improve their knowledge by referencing background information about the applied knowledge.

All information derived by the knowledge management system has to be documented for proper auditing, tracking, and provenance; data, knowledge, and derived information have to be lined up in time.

Such an application could provide a better foundation for evidence-based medicine.

Today, knowledge is shared by writing papers. There are just too many papers to read and the information is often not precise enough. Papers have to be complemented by a representation in machine useable form. This representation – with the proper precaution and adjustment – should be easily importable into a knowledge base.

Models – applied to a large number of patients - should be used to provide insight in the selection of the best 'standard of care' even taking patients' objectives into account.

3 THE TECHNOLOGIES

The development of new medical applications needs to leverage these three core technologies: data management, knowledge management, and consumer management.

3.1 Data Management

3.1.1 Data Models and Extensibility

Medical applications require the management of many types of data, far beyond classical SQL92, which is used by most applications and only supports simple structured relational data. Modern DBMSs have evolved to a point that they can store, manage, and query all kinds of data in one system, including DICOM data that are specifically designed for medical applications. Table 1 shows a few types of data that healthcare applications need with the corresponding support in modern DBMSs.

Storing data using these technologies without any intervening application logic is the preferred way of managing data. It provides an unobstructed

Table 1: Use of data types.

Data Model with Query language supported by modern DBMS	Data instance examples in Healthcare Application
SQL92 relational data with SQL	Patient id, name, birth-date etc
XML with XQuery & SQL/XML	Doctor notes, comments with XML tagging
RDF/OWL with SPARQL	Knowledge management
Domain specific data with object SQL and extensibility	DICOM patient data

view at data.These technologies are typically built on top of the database extensibility framework (Stonebraker et al., 1998).

3.1.2 Data Mining and (Online) Scoring

Information can be extracted from records by defining expressions (rules); however, what can be done to extract complex information, i.e., I need to assess the likelihood of one or more patients having a cardiac arrest within the next 24 hours? It is close to impossible to formulate the right expressions (rules) manually; data mining can be leveraged for the development of (non-hypothesis) models as well as scoring these models based on patient information. Modern DBMS can support data mining efficiently (Milenova et al., 2005).

3.1.3 History Management

Capturing the history of specific patient data with program logic is a very resource intensive in terms of development effort (typically > 50% of this effort) as well as storage and access cost. Transactional temporal databases, i.e., databases that keep previous and deleted versions of records and make them easily accessible are a much better approach. Additional benefits can be gained from RDBMS with full ILM (Information Life Cycle) support, including compression, access optimization, and more. Data history support provides the base for auditing, tracking, and provenance. A good example for temporal support is Total Recall (TR) in the Oracle database (Venkata et al., 2008).

3.1.4 Event Processing

Databases have significant support for event processing. Trigger and (online) log mining are widely available technologies (Chandy and Gawlick, 2007). A more recently introduced capability, is to register queries and notify applications and/or users if there is a change in an object or a result set (Oracle, 2010a). While registered queries allow extracting important information from databases in a timely fashion. Materialized view capability can handle more computational intensive extraction in periodical fashion.

3.1.5 Tagging and Grouping

Annotating information with tags is a flexible way of classifying and grouping information; e.g., *normal, guarded, serious,* and *critical* is a meaningful tag set to classify a wide range of patient data. Using this tag set one can ask for any data values that are classified as *serious*. Another important tag set is the diagnosis. These tags could be complemented by information about uncertainty, relevant data/tests to reduce any uncertainty, and other information.

Tags reduce significantly the complexity and the number of queries. This is due to the fact that a valuable part of the domain knowledge is captured in the selection of the tag attributes, the tag sets, and the calculation of tag values. While there is a wide consensus within a domain about these selections and calculations, it is important that customization for special areas, groups, and individual views is well supported and easy to use.

Tagged data can be modelled using XML data model and querying of tagged data can leverage XQuery that is designed to query XML (Liu et al., 2005) while RDF/OWL technology can be leveraged to support customization.

3.2 Knowledge Management

The success of modern patient care applications requires effective management of domain knowledge. This section explains the essence of (*operational*) knowledge lifecycle management, and how such knowledge can be applied to data and produce relevant timely information.

3.2.1 Knowledge Lifecycle

Knowledge Capture: Before knowledge is captured in the form of logical assertions, an ontological model (T-Box) must be specified, to provide the logical foundation for the assertions to be captured. Such model may initially be very simple. As *Subject Matter Experts* (SMEs) gain more knowledge about better ways of interpreting the facts in the database, the model can be evolved to support such learning. Support for personalized ontological model and assertions are essential for lowering the barrier to capturing knowledge. Community or institutional knowledge is eventually derived from the aggregated personalized knowledge. Also, allowing

SMEs to record their subjective belief in a generalized manner without restrictions, lower the barrier to capture such belief. Such generalized knowledge can be eventually specialized at a later time by subjecting them to deductive rules and machine learning techniques.

Knowledge Validation: Validating captured knowledge and assigning pedigree to its models and assertions is essential to the trustworthiness of such knowledge. It can be achieved through ontology validation heuristics, computing statistical consensus, or social network analysis. Increasing the trustworthiness and consistency of knowledge content tend to increase the knowledge base resistance to invalid or incorrect assertions and models.

Knowledge Retrieval: Knowledge retrieval is essential for sharing, transferring, and learning the knowledge of a community of practice or an SME. Database embedded RDF/OWL technology simplified, to great extent, the browsing, querying, and searching of such knowledge (Das, S. et al., 1004), (Oracle 2010b). While an NLP based search interface is not within the scope of this paper, it is essential to effective knowledge search.

Knowledge Maintenance: Knowledge maintenance is essential for the evolution of SMEs and community understanding and interpretation of the facts. The ability to specify defined classes and infer membership and subsumption of such defined classes is extremely important to evolving the personal and institutional knowledge. Such capabilities facilitate the refinement of the knowledge and enable the encoding of deeper understanding of the facts on ongoing basis.

3.2.2 Knowledge Application

In our approach, knowledge is maintained in two representations; a formal model that is platform independent and an executable model that is platform specific. Both models are required to be declaratively specified making it possible to maintain consistency between the two models, should one of them changes. As SMEs and Knowledge curator maintain and evolve the knowledge, a system would be able to reflect such evolution on the platform specific model using automatic transformation rules.

Similarly, changes to the platform specific model by IT personnel and data mining specialist can be reflected on the platform independent model. This will also allow us to test the validity and consistency of such change.

In the following section we will show that the current database technology is able to encode operational knowledge effectively and to process data transactionally in real-time using deductive reasoning, event processing pattern recognition, and meta-queries.

Deductive Reasoning: Deductive database research recognized the need for applying deductive reasoning to deduce actionable information from the database. It facilitated the application of logic and inference to relational data (Tsur, 1991) (Ramamohanaro, K. and Harland J. 1994). Modern databases (Oracle, 2010b) can support the transactional application of deductive reasoning in real-time. Semantic Assisted Queries is one example. Such capabilities have enabled the creation of new breed of applications capable of applying domain knowledge to the facts in the database to extract humanly consumable information in real-time (Universitaet Bonn, 2010).

Event Processing: Timely awareness of critical incidents – alerting - is very important in the medical and almost any other domain. Alerting requires users to specify which information is important to him/her. Database triggers represent the first generation of active data supporting explicit event processing paradigm (Chandy and Gawlick, 2007). Though trigger support is effective, it is still too low level and is not scalable with large number of triggers. Instead, second generation event processing in active data management enables user to register large number of patterns and efficiently evaluate them in a scalable manner via indexing over conditional Expression. Patterns of interests can be defined using conditional expressions

A data type that is of special interest for patterns is the expression type (Yalamanchi et al., 2003). Expressions can be stored in tables with a column of type expression. Given input data, one can find all the stored expressions that are evaluated to be true for these data. The scalable is achieved through a specific index for expressions. Expression filter are very effective in supporting personalized multi-level pattern recognition preferences. Event Processing facilitates the encoding of rules knowledge in the form of Event-Condition-Action.

Continuous Query Notification (CQN) on the other hand continuously evaluates the state of an object or a query and alert users and application in the event of occurrence of such change (Oracle, 2010a). Materialized view can be used to derive new knowledge given new patterns are found.

Meta-Queries: As a large body of (medical) knowledge is captured and encoded in declarative

rules and queries, it is important to apply such knowledge to (newly arriving) patients with a large medical history. Therefore, there is a need for queries that reference the knowledge base as well as the fact base to extract important information. We call these queries meta-queries. The result of these queries is a set of records that indicate which elements of a knowledge base identified relevant queries and what was found; e.g., a meta-query may identify a set of diagnosis, their time, and even associate probability.

It is not feasible to apply all rules to all patient records due to the length of time it takes to apply all known rules and queries to patient records. One way to find such relevant queries is to treat the rules and queries as an abstract data type, build an efficient index, and identify relevant queries by using patient data to traverse this index.

Meta-queries provide the same results as event processing if the same queries and data are used. The only difference is the time results become available.

3.3 Consumer Management

An application that supports data and knowledge management can deduce information from raw data as soon as new information becomes available. This interpretation of data, however, often depends on the personal view and preferences of individual doctors – the consumers of the information. Therefore it has to be known which doctors are responsible for which patients in order to apply the right profiles (of knowledge) to incoming data. If the deduced information is classified as being urgent doctors (or other medical personal) have to be alerted immediately. Obviously, the deduced information as well as the urgency can be different for different doctors –allowing doctors in a team to share each other's view may be beneficial.

This function requires up-to-date data about the consumers (medical personnel), their assignment, their availability, how to reach them, as well as their (evolving) profiles; i.e., the status of all consumers and their organizational context must be known at any point in time.

4 PUTTING IT ALL TOGETHER

Applications depend on four main components; knowledge, data, information, and consumers.

Knowledge Base contains the structure and semantics of the data and all the codified domain knowledge. The codified knowledge can be represented in the form of rules, expression (*if expression then …*), models or any other means useful to extract information from data. The knowledge base is highly customizable, it contains the codified knowledge of the entire medical community customized by views and interest of doctors working in specific domains, as well as customized by the view of individual doctors, and even the way doctors looking at specific patients.

Data contains all the measurements and observations that are captured by sensors and the medical personal; here we will record the vitals, the blood chemistry, and whatever else has been observed. Data supports a wide range of data types, history management, life cycle management as well as all essential operational characteristics, such as performance, scalability, availability, fault/disaster tolerance, and security.

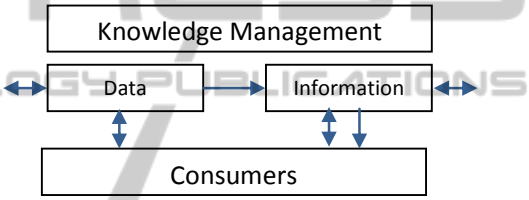

Figure 1: Application components.

Information contains insights that have been derived from applying the operational knowledge; here we will find a note if the heart beat is perceived as critical or if a patient has a high risk of a cardiac arrest within 24 hours. The underlying functional and operation support is at the same level as for data.

Consumer contains details about the medical personnel, such as who is on shift, who is available, and how to reach a doctor or nurse. Most importantly there is data on the customization of the operational knowledge.

Raw data will be captured using OLTP or protocols; export is possible as well.

Whenever new data are captured, they and the previously captured data will be analyzed using the operational knowledge, if anything of importance is found, one or more entries in information base will be made; e.g., data will be transformed into information. It should be noted that information entries could be created even if there is no change in data. For example, the blood pressure has remained high for an extended period. Doctors can add informational notes as well.

Entries in the information base may be tagged as time critical; so as to alert attending doctors/nurses.

In addition to being alerted, the medical personnel can access data and information entries at

any point in time. If a doctor gets an alert he/she may be interested in the background leading to this alert, the system will be able, not only to show which data triggered the alert, but also which entries in the knowledge base were used. This is especially helpful if an alert is triggered by a knowledge base entry representing medical knowledge a doctor is not familiar with.

Obviously, doctors can ask questions referring to information such as *what is important to know about this patient*; rather than going through 1000's of EMRs.

5 COMPARISON

It may be interesting to compare our approach to OLTP and (stream based) event processing. OLTP only manages data; there is neither a knowledge base, nor information, nor consumer management. Event processing supports the data component only in a very specific way; entries in data are transient and ordered by a time stamp, there is no query support and the focus for operational characteristics is only on performance. Knowledge management is implicitly represented as a set of rules or continuous queries, results from queries/rules are *thrown over the wall*; there is no information and consumer management. Our approach is integrated and it is based on deductive database concepts enhanced by tagging, grouping, and consumer management. This direction is aligned with the work in (Universitaet Bonn, 2010).

6 CONCLUSIONS

The ISCU prototype has shown that it is possible to develop a patient care application that can be universally used in ICU environments and other patient care departments in hospitals, institutions, out-patient care, as well as long term home care. This application manages data, knowledge, information, and consumers. It captures data, deduces information from these data based on personalized knowledge, alerts if urgent actions are required, and provides unobstructed access to data, information, and knowledge.

Our approach of integrating data, information, knowledge, consumer management enables medical applications to be developed in declarative manner so that domain experts, such as medical staff, can share knowledge and experience – easily, flexibly with little delay. Such application can evolve without dependency on IT personal and adjust to advances in IT technology.

ACKNOWLEDGEMENTS

We like to thank Ute Gawlick (MD/PhD) for ensuring that the SICU prototype reflects the needs and thinking of the medical community, Diogo Guerra for developing a complex prototype in only 4 months, and Pablo Tamaya for developing a non-hypothesis driven predictive model for cardiac arrest. Their work is the base for this paper.

REFERENCES

Chandy, K., Gawlick D., 2007. Event processing using database technology. SIGMOD Conference 2007: 1169-1170.

Das S., Chong E., Eadon G., Srinivasan, J., 2004. Supporting Ontology-Based Semantic matching in RDBMS. VLDB 2004: 1054-1065.

Guerra, D., Gawlick, U., Bizarro P., 2009. An integrated data management approach to manage health care data. DEBS 2009

Liu Z., Krishnaprasad, M., Arora, V., 2005. Native Xquery processing in Oracle XMLDB. SIGMOD Conference 2005: 828-833.

Milenova B., Yarmus, J., and Campos, M., 2005. SVM in Oracle Database 10g: Removing the Barriers to Widespread Adoption of Support Vector Machines. VLDB 2005: 1152-1163.

Motro, A., Yuan, Q., 1990. Querying Database Knowledge, ACM 069791 365 5/90/0005/0173.

Oracle, 2010a. Using Continuous Query Notification, http://download.oracle.com/docs/cd/E11882_01/appdev.112/e17125/adfns_cqn.htm#CHDIEHHJ.

Oracle, 2010b. Semantic Technologies, http://download.oracle.com/docs/cd/E11882_01/appdev.112/e11828/toc.htm.

Ramamohanaro, K. and Harland J., 1994. An Introduction to Deductive Database Languages and Systems, VLDB Journal, 3, 107-122.

Stonebraker, M., Moore D., Brown P., 1998. Object Relational DBMSs: Tracking the Next Great Wave, 2nd edition. Publisher Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Tsur, S., 1991, Deductive Databases in Action, ACM 0-89791-430-9/91/0005/0142.

Universitaet Bonn, 2010. Active and Deductive Database, http://idb.informatik.uni-bonn.de/research.

Venkata, K. Kanth, R., Hanckel R., Yalamanchi, A., 2008. Using Oracle Extensibility Framework for Supporting Temporal and Spatio-Temporal Applications. TIME 2008: 15-18.

Yalamanchi A., Srinivasan J., Gawlick, D., 2003. Managing Expressions as Data in Relational Database Systems. CIDR 2003.