KIDS
A Model for Developing Evolutionary Database Applications

Zhen Hua Liu[1], Andreas Behrend[2], Eric Chan[1], Dieter Gawlick[1] and Adel Ghoneimy[1]
[1]*500 Oracle ParkWay, Redwood Shores, CA 94065, U.S.A*
[2]*Universität Bonn, Institut für Informatik III, Römerstr. 164, 53117 Bonn, Germany*

Keywords: Knowledge Management, Information, Data Management, Event Processing, Workflow.

Abstract: Database applications enable users to handle the ever-increasing amount and complexity of data, knowledge as well as the dissemination of information to ensure timely response to critical events. However, the very process of human problem solving, which requires understanding and tracking the evolution of data, knowledge, and events, is still handled mostly by human and not by databases and their applications. In this position paper, we propose KIDS as a model that reflects the way human are solving problems. We propose to use KIDS as a blueprint to extend database technologies to manage data, knowledge, directives and events in a coherent, self-evolving way. Our proposal is based on our experience of building database centric applications that require comprehensive interactions among facts, information, events, and knowledge.

1 INTRODUCTION

Current database technologies are supporting many modern mission critical applications. There are two major reasons: a declarative interface and a wide range of important operational characteristics. The declarative interface supports queries and data manipulations without the need to understand implementation details. The operational characteristics provide the support for OLTP applications that serve many users concurrently, high security, and high availability with no loss of data or service. With parallel query processing, data partitioning, data mining, business intelligence, and decision support databases can analyze large amount of data, recognize patterns, and extract useful information from raw data upon which intelligent decisions can be made. Active DB technologies, such as triggers, rule processing, continuous execution of registered queries, and streaming query processing, can be used to recognize and monitor interesting events and secure timely responses .

With modern databases individuals and institutions can solve problems by collecting and storing factual data, using knowledge to analyze such facts to develop directives for follow up activities, and finally, performing such directives and capturing their effects in the form of additional facts for further analysis. Such analysis either indicates that the problem has been resolved (goal achieved) or further analysis is required. This cycle of inspecting results, specifying directives and acting upon them will be repeated until the problem is resolved. Although modern DB based applications help human at every step of the process, it is human, not the database system itself, who understand, track and follow up the process. The human mind can handle simple problems without support. On the other hand, complex problems requiring many repeated iterations of the process with large amount of evolving data, knowledge directives, and the collaboration of teams, require an infrastructure that tracks problem solving activities in a scalable manner. As a side effect, the effective management of generated facts, information, and directives can lead to enriching the existing knowledgebase.

Since problem solving relies heavily on tracking the state of facts, knowledge, information, and activities, we believe that database technologies provide the right foundation since their traditional focus and strength is on providing *scalable state tracking and management service in a declarative fashion*. In this position paper, we propose enhancements to database technologies to create a comprehensive problem solving platform. We propose the KIDS model for describing the life cycle of data management tasks. The acronym KIDS stands for the most important elements of this model

Hua Liu Z., Behrend A., Chan E., Gawlick D. and Ghoneimy A..
KIDS - A Model for Developing Evolutionary Database Applications.
DOI: 10.5220/0004002501290134
In *Proceedings of the International Conference on Data Technologies and Applications* (DATA-2012), pages 129-134
ISBN: 978-989-8565-18-1

by means of **K**nowledge, **I**nformation, factual **D**ata, **D**irectives and **S**ervices. The interrelationships among the elements of KIDS are presented in Figure 1.

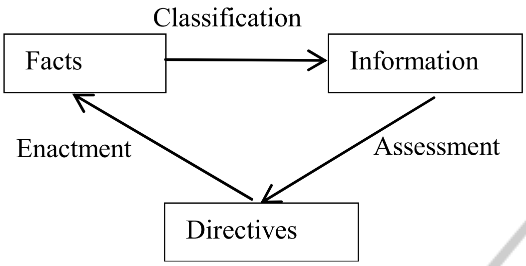

Figure 1: High Level KIDS Model.

KIDS distinguishes among three classes of **data** (*facts*, *information*, and *directives*) and three classes of **knowledge** (*classification*, *assessment*, and *enactment*). Solving problems entails the capturing and the reduction of emerging and historical facts into information by applying classification knowledge. Then such information is used to assess the situation and prescribe/describe the directives for dealing with the situation. Finally, the directives have to be executed by applying enactment knowledge. As directives are enacted, newly emerging facts will again be classified to determine whether the situation has been resolved or not. By implementing KIDS as a database centric platform, we will not only enable applications to function as an eco-system that can track the evolution of data, knowledge, directive that facilitate effective problem solving processes, but also make the entire application itself query-able with time traversal service, "what if" analytical service and provenance tracking using modern DB temporal and provenance technologies.

Techniques of modeling data and process logic are not new. EEML (Krogstie, 2008) provides modeling languages to express data, process, resource and goal modeling logic. However, our focus is to integrate data and process tracking deeply into DB technologies whose strength is to provide scalable and declarative state tracking service with time and provenance support. To our best knowledge, this is the first position paper that makes the following contributions:

- This paper proposes the KIDS model and its underlying concepts and shows how to leverage the strength of DB technologies to provide KIDS as a database centric service.
- This paper derives a blueprint to organize and extend DB technologies, based on the challenges encountered in supporting a database centric KIDS platform.

The remainder of the paper is structured as follows: Section 2 describes in detail the major use case – patient care that motivates KIDS, section 3 describes KIDS concepts in detail, section 4 is a functional illustration of how KIDS can be supported using Object SQL, section 5 discusses the challenges of supporting KIDS in DB, and section 6 discusses proof points with conclusion in section 7.

2 PATIENT CARE USE

Health care providers are confronted with an increasing amount and complexity of data and knowledge as well as an increasing amount of rules and regulations.

The capturing of patient data as EMRs (Electronic Medical Records) is becoming common place. The management of EMRs requires storing a wide variety of data without any loss while providing instant access at any time as well as comprehensive security.

The review and the interpretation of medical data is becoming increasingly time consuming and controversial. Therefore, modern patient care applications have to handle such challenge; i.e., doctors need a system that transforms EMRs into compact information, applying the codified medical knowledge and providing all possible interpretations. This must be done on demand as well as proactively in real time to alert doctors and nurses about adverse and time critical situations. The number of false positive and negatives has to be kept as low as possible to avoid the alert fatigue that is observed with practically any existing alerting systems.

The system should leverage any encoded medical knowledge to reduce vitals, the blood chemistry, the radiology, other observation, or any mix thereof into one or a few classes of patient state and alert doctors if such classification indicates the presence of one or more possible adverse situations that need attention.

This transformation has to be well documented, and should be easily personalizeable to the preference of each doctor while observing the institution's constraints.

Once the doctor is alerted of the situation and supplied with the information summarizing the patient condition along with the relevant facts, s/he assesses the situation and decides on the course of action, including any changes to existing treatment. The application should guide doctors using the standard of care, models, or any other means to advice about the best course of action; e.g., indicate which medicine or combination of medicines has been must successful with patients in a similar situation and also which tests are advisable to reduce the level of uncertainty of the diagnosis. Once the orders are submitted, the system needs to help in the supervision and documentation of the execution.

In essence, doctors need comprehensive support in all phases of the treatment including: the capturing of the facts (EMRs), the extraction of information from these facts, the assessment of the relevant information and facts (diagnosis), determine the course of action (directives - orders), and the enactment of such directives. The system should allow doctors to personalize the application according to their preferences, be able to integrate new knowledge easily, contribute to the development of knowledge, and still leave any crucial decision in the hand of the doctors.

3 KIDS CONCEPTS

KIDS provides three base classes: the Data, Knowledge, and Actor classes. An in-depth discussion of the Actor classes will be covered in a different paper.

3.1 The Three Classes of Data

Data is factored into Facts, Information, and Directives that are defined as follows:

Facts represent anything that is observed and captured; examples are the stream of data coming from monitors in an ICU, a doctor's observation, and an x-ray or ultrasound. Facts provide a **quantitative** representation of what has been observed.

Information represents an interpretation of the facts; examples are **qualitative** values assigned to a vital (the blood pressure is normal, severe, or critical) or the assignment of a name to a pattern – a diagnosis. Information represent are a qualitative interpretation of what has been observed.

Directives represent a description of activities to be carried out; examples are prescriptions for medicine and orders to test certain aspects of the blood chemistry. Directives move the task on hand ahead and trigger the acquisition of new facts.

3.2 The Three Classes of Knowledge

The three classes of data are handled by corresponding three classes of knowledge: Classification, Assessment, and Enactment. These three classes of knowledge provide the necessary ingredients for abstracting most activities of any medical, scientific, or business institution. The definitions of knowledge classes are as follows:

Classification knowledge reduces newly arrived quantitative facts into qualitative information. An example is the association of a severity to a vital, such as the blood pressure, is critical based on the blood pressure measurement and the general condition of the patient. Another example is the identification of diseases based on the available observation. The goal of the classification is to derive a compact representation of important facts.

Assessment knowledge analyzes the new information to formulate a hypothesis about the root cause and finally render a set of directives to respond to the situation. An example is the selection of medicine to treat a patient in which the case that the directives are the prescriptions. This assessment could be purely manual or computer guided; e.g., using models that show the best selection for patients in a similar situation. The goal of the assessment is to development the best directives based on goal.

Enactment knowledge interprets directives and carries out their intent. Examples would be the application of medicine, a single or repeated blood test, the activation or change of health monitors, and a surgery. The goal of an enactment is to solve a problem, and/or capture additional facts.

3.3 Evolution with FID Loop

The three types of knowledge support the evolution of facts, information, and directives; we call this the FID loop. The FID loop is the '*heart beat*' to provide live process control of KIDS concepts. With time and provenance tracking, the evolution of the data and the knowledge in a FID loop can be completely reviewed, queried, analyzed for auditing, for "what if" analysis, for knowledge enhancements etc.

4 KIDS SERVICE FROM DATABASE SYSTEMS

Databases shall provide services to allow any KIDS instance to be stored and accessed declaratively. Additionally, databases shall keep every version of each instance in support of temporal and provenance access. By storing multiple instances of KIDS, many users shall share and work the same or multiple KIDS instances collaboratively. Whenever knowledge gets added, changed, or deleted, the existing data elements will be processed again. This ensures that the most current knowledge is always applied to the most current data.

In this section, we show how KIDS concepts can be accessed declaratively using OR-SQL.

```
/* This creates an object table serving as a container to hold all
 KIDS object instances */
CREATE TABLE MYCLINICS OF KIDS_TYPE;
/*now populating two KIDS instances into the table containers */
/* creation of KIDS instance for patient 'John Smith' */
INSERT INTO MYCLINICS VALUE(
KIDS_TYPE('John Smith Case',
FACTS_TYPE(SELECT PATIENT_OBJ
      FROM PATIENTS
      WHERE PATIENT_NAME = 'John Smith'),
INFORMATION_TYPE(BLOOD_CHEM_REPORT_VIEW,
CARDIOLOGY_REPORT_VIEW),
DIRECTIVE_TYPE(ORDER_BLOOD_PRESSURE_CONTROL_PROC)
as d1,
KNOWLEDGE_CLASSIFICATION_TYPE(BLOOD_CHEM_ANA_FUN
C, CARDIOLOGY_ANA_FUNC, BLOOD_PRESSURE_ANA_FUNC),
KNOWLEDGE_ASSESSMENT_TYPE(WHEN
BLOOD_PRESSURE_ANA_FUNC() = "HIGH" EXEC d1));
/* creation of KIDS instance for patient 'Mary Dunn' */
INSERT INTO MYCLINICS VALUE(
KIDS_TYPE('Mary Dunn Case',
FACTS_TYPE(SELECT PATIENT_DOC
      FROM PATIENT_DOCS
      WHERE PATIENT_NAME = 'Mary Dunn),
INFORMATION_TYPE(BREAST_CANCER_PROGRESS_VIEW),
DIRECTIVE_TYPE(ORDER_CHEMO_PROC) as d1,
KNOWLEDGE_CLASSIFICATION_TYPE(BRST_CANCER_ANA_FUN
C),
KNOWLEDGE_ENACTMENT_TYPE(BRST_BIOPSY,SURG_REMOVA
L_PROC p1)
KNOWLEDGE_ASSESSMENT_TYPE(WHEN p1.status = 'done' EXEC
d1));
```

SQL Code 1: KIDS container and object instances.

SQL code 1 illustrates the creation of table to store KIDS object type instances.

```
SELECT OBJECT.GET_FID_LOOP('PROVENANCE')
FROM MYCLINICS
WHERE OBJECT.DESCRIPTION = 'Marry Dunn Case'
```

SQLCode 2: Provenance Query.

Provenance Retrieval: *SQLCode 2* retrieves the provenance for patient '*Marry Dunn*' by calling the method *GET_FID_LOOP()* in '*PROVENANCE*' mode. This method shows the entire FID loop execution sequence of treatments for the respective patient. This includes data and knowledge classification rules that are used to derive Mary's cancer progress report, e.g. which surgery and chemo-therapy have been applied. Such provenance output allows doctors to examine and navigate the FID loop process so that they can judge the trustworthiness of the diagnosis and treatment plan information. Provenance computation relies on the full version history of the underlying KIDS components. It supports *time traversal* aspect of KIDS.

Knowledge Evolution: Knowledge evolves over time. Doctor may be interested, in the case of patient '*John Smith,*' to use the latest medical knowledge advancement for blood chemistry analysis. *SQLCode 3* shows how this can be accomplished using a conditional update statement. First, in a workspace, the update statement is employed to set the knowledge classification rule to use the latest version of blood chemistry analysis function for patient '*John Smith*'. Afterwards, it calls the method *GET_FID_LOOP()* using the mode '*PROJECTED*'. In this way, unchangeable historical data can be reprocessed using "what if" kind of analysis.

```
BEGIN WORKSPACE;
UPDATE MYCLINICS
SET OBJECT.KNOWLEDGE_CLASSIFICATION
= BLOOD_CHEM_ANA_FUNC(LATEST)
WHERE OBJECT.DESCRIPTION = 'John Smith Case';
SELECT OBJECT.GET_FID_LOOP('PROJECTED')
FROM MYCLINICS
WHERE OBJECT.DESCRIPTION = 'John Smith Case';
END WORKSPACE;
```

SQLCode 3: Knowledge Evolution Analysis

Personalization and Collaboration: Knowledge application is not always absolute; e.g., different doctors may derive different information from the interpretation of the same data. KIDS shall allow users to customize knowledge application to their preferences. That is, KIDS is personal context aware. Furthermore, KIDS shall promote user collaboration. With permission, multiple doctors shall be able to see each other's contexts and the derived information and directive due to these contexts. This facilitates knowledge collaboration among doctors. *SQLCode 4* shows how this can be accomplished declaratively by calling the method *GET_FID_LOOP()* and passing the '*PROJECTED*' mode with Dr. *Ute Gawlick*'s knowledge.

```
BEGIN WORKSPACE;
UPDATE MYCLINICS
SET OBJECT.KNOWLEDGE = (SELECT KNOWLEDGE
        FROM KOWLEDGETAB
        WHERE
        DOCTOR_NAME='Ute Gawlick')
WHERE OBJECT.DESCRIPTION = 'John Smith Case';
SELECT OBJECT.GET_FID_LOOP('PROJECTED')
FROM MYCLINICS
WHERE OBJECT.DESCRIPTION = 'John Smith Case';
END WORKSPACE ;
```

SQLCode 4: Knowledge Personalization & Collaboration Query.

5 DATABASE CHALLENGES TO PROVIDE KIDS SERVICE

This section illustrates the challenges of hosting KIDS service from DB perspective.

Knowledge as First Class Citizen: Modern RDBMSs already support concepts from expert and knowledge-based systems by means of RDF, OWL and further logical reasoning capabilities (Das, 2009). Additionally, machine learning techniques have been incorporated into modern RDBMSs in the form of data mining functions (Agrawal et al, 1994), (Milenova et al, 2005). However, compared with the declarative way of querying data, database systems are still weak in terms of providing the same support for *querying knowledge*. In particular, a user cannot declaratively query application knowledge coded in view definitions, stored procedures, triggers, or event- processing handlers. Being able to classify, query, search, browse and validate knowledge is necessary to make knowledge a first class citizen of a RDBMS just like data is. To this end, all forms of knowledge, whether represented in form of inference rules, statistical classifications, learning algorithms, conditional expressions, query qualifications, or procedural code, ought to be indexable and its modification should be automatically monitored and version tracked as if they were plain data.

Active Knowledge Application: Classical database systems require users to play an active role applying knowledge to facts. In contrast, active knowledge application refers to applications with knowledge actively looking for facts that are needed to achieve the actor's goals. This is done by monitoring fact updates. Additionally, registered queries and real-time scoring can be used for automatically deriving new information from evolving facts. However, answering questions like what knowledge or facts are missing or needed to be changed in order to derive certain information and to execute certain directives to fulfill actor's goal require abductive reasoning techniques (Denecker et al, 2002) which are not yet available in database systems.

Full Version History and Provenance Awareness: Data in the form of facts, information, and directives, as well as knowledge in the form of classification, assessment, and enactment are intrinsically temporal and thus need to be versioned tracked. KIDS service requires temporal database support with snapshot isolation to access consistent versions of data and knowledge to extract provenance. Multi-version index structures (Becker et al, 2005) with declarative temporal expressions are necessary for efficiently tracking the development of data and knowledge over time. Although data provenance is already a research topic (Karvounarakis et al, 2010), the integration of provenance with workflow and process management is still a challenge. Such general form of provenance enables users to navigate within the FID control loop and examine the actual instances of data and knowledge that have been used at each loop step. In this way, it is feasible to provide time traversal of KIDS instances so that user can understand how historically conclusions were reached and decisions were made. Furthermore, although history is not alterable, it is feasible to do "what if" analysis by generating a new branch of history with application of latest knowledge to historically collected facts.

Quality Control and Collaboration for KIDS: KIDS instances have to enable groups of people to help and learn from each other. Users shall be able to share parts of a FID loop enabling other users to be engaged. This type of collaboration approach supported by version tracking and provenance allows for improving the quality of data and knowledge (Richardson et al, 2003).

6 PROOF POINTS

The KIDS concepts have been used for guiding the implementation of a patient care prototype for an SICU (Surgical Intensive Care Unit) at the University of Utah (Guerra et al, 2011). The prototype consists of a single data repository that combines a highly configurable rule-based system; (push-based) alerts, data mining models and an intuitive user interface. Everything is highly customizable to the preferences of doctors and the specific circumstances of patients. The prototype is able to predict life-threatening events (e.g., cardiac arrest).

The ideas of this prototype have the potential to significantly increase the value of healthcare information as the database can store and analyze healthcare records continuously, find critical situations, explain why they are critical, allow for a detailed investigation and provide recommendations - even for diseases or critical situations that a specific doctor has never encountered or does not even know about. An important side effect of this approach is that the number of false positive and negative alerts has been significantly reduced; reducing significantly the familiar alert fatigue. This approach has the potential to save lives and improve health care not only in ICU settings by also for any inpatient and outpatient services.

Another subset of KIDS concepts concerning knowledge elicitation, structuring, maintenance, and evolution, has been empirically proven for troubleshooting and resolving software and hardware issues. A case-based approach to knowledge elicitation from subject matter expert was a key ingredient to the success of our knowledge elicitation method. The agile and economic maintenance of such models is facilitated by case-based automated regression testing framework. Finally, we found out that leveraging models based on structured knowledge is more effective when the information to such models is evaluated automatically, rather than relying on extracting such information from users.

7 CONCLUSIONS

In this position paper, we have illustrated the KIDS concepts and FID loop that models human problem solving process. Since individual pieces of KIDS concepts, such as fact, information, directives and knowledge, have already been managed by DB system, it is time to use KIDS as a blueprint to extend DB technologies to understand and manage the whole KIDS FID loop process and to provide declarative KIDS service so as to assist human problem solving in a scalable manner. The challenge is ahead of us: deep integration of knowledge management and process management into DB database technologies are essential to the support of KIDS abstractions. It is our vision that KIDS enabled database technology shall be able to host applications that track evolution of data, knowledge, directives, events, in a scalable, consistent and self-evolving manner.

REFERENCES

M. Richardson, P. Domingos: Building large knowledge bases by mass collaboration. *K-CAP* 2003: 129-137

R. Agrawal, R. Srikant: Fast Algorithms for Mining Association Rules in Large Databases. *VLDB* 1994: 487-499

D. Guerra, U. Gawlick, P. Bizarro, D. Gawlick: An Integrated Data Management Approach to Manage Health Care Data. *BTW* 2011:596-605

G. Karvounarakis, Z. G. Ives, V.Tannen: Querying data provenance. *SIGMOD Conference* 2010: 951-962

E. N. Hanson, C. Carnes, L. Huang, M. Konyala, L. Noronha, S. Parthasarathy, J. B. Park, A. Vernon: Scalable Trigger Processing. *ICDE* 1999: 266-275

S. Das, J. Srinivasan: Database Technologies for RDF. *Reasoning Web* 2009: 205-221

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

Bruno Becker, Stephan Gschwind, Thomas Ohler, Bernhard Seeger, Peter Widmayer: An Asymptotically Optimal Multiversion B-Tree. *VLDB* J. 5(4): 264-275(1996)

M. Denecker, A. C. Kakas: Abduction in Logic Programming. *Computational Logic: Logic Programming and Beyond* 2002: 402-436

J. Krogstie: Using EEML for Combined Goal and Process Oriented Modelling: A Case Study: *EMMSAD* 2008