A KNOWLEDGE OBJECT ORIENTED SYSTEM FOR HIGH THROUGHPUT COLLECTION AND ANALYSIS OF DATA

Huiqing Liu
BioInformatics Centre, National University of Singapore
Emai: huiqing@bic.nus.edu.sg

Tecksin Lim
KOOPrime Pte Ltd, 71A Tanjong Pagar Road, Singapore
Email: *tecksin@KOOPrime.com*

Keywords: Knowledge object oriented programming, high throughput data collection and analysis, business logic management

Abstract: An approach, KOOP (Knowledge Object Oriented Programming) that can integrate, personalize and automate a set of processes based on the business logic has been developed. This technology processes high throughput data in heterogeneous and distributed environment. In addition, KOOP has the ability to trap the explicit, implicit and hidden knowledge embedded in the business flows.

1. INTRODUCTION

With the growing interest in areas like genomics and proteomics, and the advent of high throughput technologies like DNA Microarrays, life science users are increasingly being overwhelmed by the huge amount of data generated. To satisfy the growing need to organize and analyze the data, a number of key technological areas are identified and focused upon, namely *Business Logic Management* (BLM), *High Throughput Data Collection and Data Analysis* (HTCA). An integrated system that can meet the needs in these areas will not only facilitate large volume data collection and analysis efforts, but also manage the implicit process knowledge involved in the business flows.

The concept of KOOP was developed in the BioInformatics Centre of National University of Singapore (NUS) and Centre for Natural Products Research (Singapore) over a period of 4 years under BioMining project that was funded by NUS and GlaxoWellcome (now GlaxoSmithKline). KOOP stands for *Knowledge Object Oriented Programming*, an approach for building systems that can evolve with business processes. It is a new paradigm that is built on top of the popular OOP (Object Oriented Programming) approach (Lim, 2000).

2. KOOP

Similar to the concept of OOP, KOOP aims to develop systems as modular and reusable objects. KOOP is developed with the realisation that no framework is provided by OOP to systematically capture the interactions among objects. Objects written by different developers are not standardised and additional coding are required to integrate the objects as a complete application. This results in applications that are not easily maintainable by end users themselves. Users have to conform to the ways the developers implemented the systems and developer resources become a bottleneck. This can be an issue, especially when the environments are dynamic and software has to evolve quickly to adapt. In the life science environment, the problem is compounded by the fact that objects are often distributed in nature and users have to access them as a cohesive application.

Accepted for oral presentation at ICEIS 2002 International Conference on Enterprise Information Systems

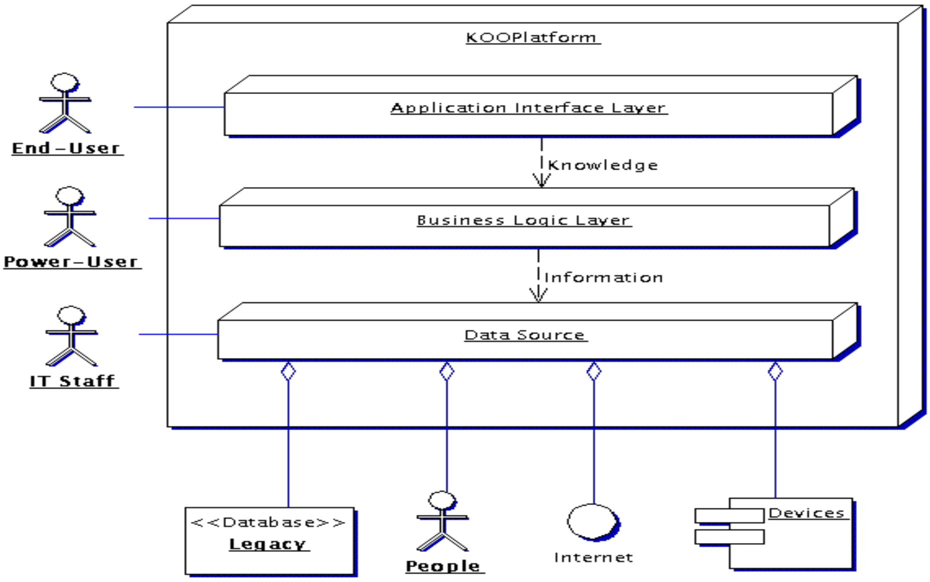

Figure 1: A high level architecture for the deployment of KOOP.

The KOOP approach is adopted to resolve the above difficulties. The first step is to develop applications as collections of normalized objects (known as "*Bubbles*") with inputs and outputs that are exposed in a standardized manner. A software platform (KOOPlatform) is then provided for user to visually personalize parameters of each bubble as a "*Knowledge-Object*" which can then be linked up with implicit knowledge (known as "*Links*"). Implicit knowledge is know-how that resides within the users and that is not shared/published. Such implicit knowledge is efficiently captured and visualised when the users personalise the bubbles and create the links between the bubbles to map the real world processes. The implicit knowledge that is saved as "bubble-and-link" templates (known as *KOOPs*) can then be maintained in several ways. Figure 1 shows a high level architecture for the deployment of KOOP in an organization. Of importance are the power users (group leaders, scientists) who are familiar with the domain knowledge and have the right to control business flows. The power users build the templates and share knowledge with their team members (end-user) who are required to execute the business processes as according to the business rules (templates) set up. The in-house IT staffs are tasked to maintain the related data sources. They maybe involved in the development of bubbles for the power users as well. Such systematic division of labour is made feasible via the KOOPlatform which aims to be a truly "open" system that can interface existing legacy software tools, databases, hardware systems and appliances.

2.1 Components of KOOPLATFORM

There are 3 main components of KOOPlatform, namely KOOPServer, KOOPBox and KOOPeer. As the front-end of KOOPlatform, the KOOPBox is a user-friendly system for both end user and power user to manage the business logic. User runs KOOPBox to design business flows via selecting (drag-and-drop) of bubbles, setting up the parameters of each bubble and linking up the bubbles into an application (i.e. KOOP template). The template that is personalized by the user can be submitted to the KOOPServer for execution, either immediately or at a scheduled time.

While the KOOPBox and KOOPServer adopt a client-server architecture, KOOPlatform is agent based as well. KOOP executes the processes in a distributed manner on remote machines via software agents known as KOOPeers. Once a KOOPeer is connected with the KOOPServer, it will listen for instructions that are sent from the KOOPServer. KOOPeer then performs the relevant task and returns results to KOOPServer. The KOOPServer checks the KOOP template to redirect the results to other KOOPeers accordingly.

A PC machine can be tasked to run the KOOPServer, KOOPBox and KOOPeer singly or all together. These systems are developed in Java with JDBC for database access, RMI and/or ORB for communication between the systems and with applications.

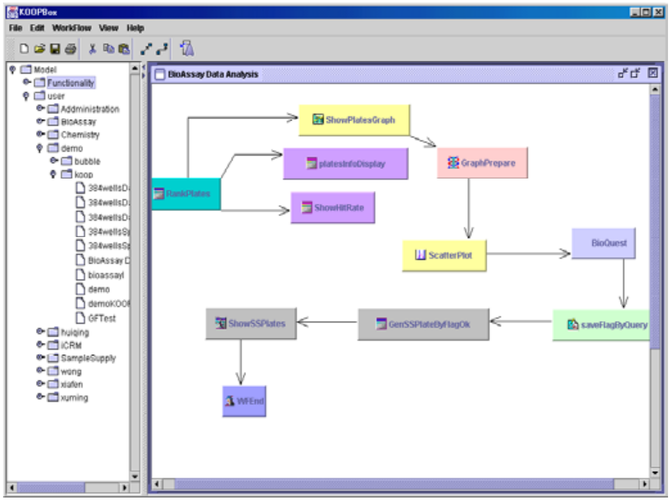

Figure 2: An Implementation of KOOPBox.

2.2 Bottom-up and top-down problem solving

The KOOP framework allows both bottom-up and top-down problem solving. It facilitates bottom-up approach because

- The modular bubbles act as building blocks for hierarchical and systematic modelling of business flows. Various generic bubbles can be used across multiple domains, such as calculator bubble, graphic bubble and etc.
- Applications are strategically modularized as bubbles to enhance code reusability. A bubble can be used in different KOOPs by same or different users.
- The bubbles can be organized via different dimensions to facilitate ease of business flow building. For example, in Figure 2, the left panel of the KOOPBox provides a visual way to categorize various bubbles and KOOPs according to the functionality and user group.
- Set of connectors can be made available for ease of bubble linkage. In KOOP, a bubble can pass its output to connected either directly or by changing to other format that the subsequent bubble require, e.g. data type changing (from an Integer to a String), from file address to file content and etc. The conversion will be done automatically by the system when necessary.
- New applications can be readily absorbed as black-box sort of bubbles to enhance existing business flows. A wizard is provided to help this process.

It also facilitates top-down solving whereby

- Business flows developed can be easily tailored and created by manipulation of the bubbles and/or subset of business flows. It is as simple as a “copy-and-paste” operation in the KOOPBox environment.
- Procedural and process knowledge can be trapped via the links created visually. This is the main advantage of using KOOP. Next section will discuss this further.

2.3 KOOP and HTCA

With the above framework, it becomes feasible to use KOOP to provide enterprise wide systems that are powerful enough to handle most HTCA domains. Although complex, such domains have just three important entities that need to be considered, i.e. human, software and hardware which are frequently heterogeneous and distributed. Figure 3 shows the three entities in the HTCA domain.

- Human entity is heterogeneous because there are different groups of users in an enterprise, all of which have their own objectives and preferences in resolving problems. The human entity could be geographically distributed.
- Software entity is heterogeneous because there are different operating systems in the market and it can be difficult to interface applications that run on these disparate platforms. The software entity is also distributed because applications can be installed on more than a machine and it may be necessary to control them as a single application.

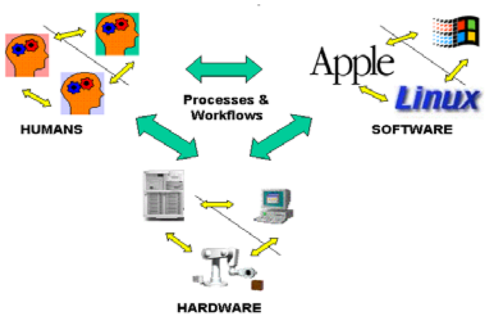

Figure 3: Handing HTCA domains via processes and business workflows.

- Hardware entity is heterogeneous and distributed because there are standalone PCs, servers and embedded systems that are spread out across different departments of the enterprise.

Most conventional systems require clients to interact with all these entities with data and knowledge using set of user interfaces and applications. Unlike these systems, the KOOP approach is to organize data and process knowledge systematically as bubbles and links such that complexity of the entities are hidden and users themselves can easily interface with all the software and hardware involved using one common user interface. Figure 3 shows how the HTCA issues can be handled via processes and business workflows that KOOP provides.

3. KNOWLEDGE CAPTURE IN KOOP

KOOP traps the implicit knowledge involved in the process flows. This is possible because the KOOPBox and KOOPServer are business logic management systems that can memorize the instructions of user. Effectively, the sequence of a set of processes, the relationship among them, i.e. the business logic, the parameters set to run a process (such as necessary inputs of the application like the hyperlinks and documents, starting time and maximum time to run the process) are tracked by the system. Once the user completes the modeling of a process via KOOPBox, he can save the KOOP template to the database for others to access. All such information can be shared by the colleagues and retrieved even if the creator of the template leaves the organization.

At the same time, KOOPServer is able to log important data during the running of processes. There are several kinds of information the KOOPServer tracks: 1) starting time, ending time and running time of a process which allow the user to identify the most time-consuming processes so that the subsequent optimization of the processes could be conducted. 2) meta-data of KOOP, which includes the latest running status and the results that are generated. Thus, it is easy to recover a KOOP process if it is to be aborted for any reasons. Instead of re-starting from the beginning process, an aborted KOOP can be re-executed from the interrupted process without any data lost. This is significant when the business flow contains many time-consuming processes. 3) login information of users. Understanding uers' profile and schedule may help balance the workload of multiple KOOPServers.

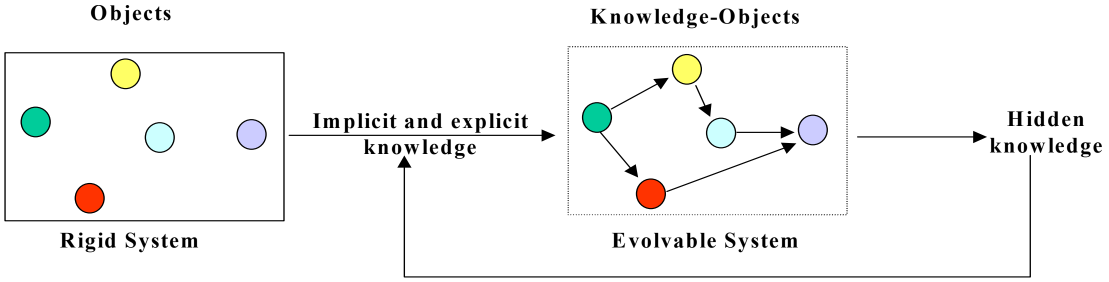

Figure 4: Management of implicit, explicit and hidden knowledge.

In order to detect hidden knowledge, bubbles that perform data mining have been integrated into the KOOP environment. More illustration is provided in the next section on the application of data mining technologies to the data stored in

KOOP. Figure 4 shows the possible management of implicit, explicit and hidden knowledge in KOOP.

4. APPLICATION

Since the technology and the software developed under KOOP are flexible and horizontal, the potential applications are huge. The technology can be applied to many domains, including biotechnology, financial, healthcare, supply chain, etc. The initial focus has been set to provide business intelligence for biological databases and specialized enterprise information portals such as intelligent Customers Relationship Managementand Pre-clinical trial solutions for healthcare purposes.

One of the test-beds of KOOP technology involves a center that performs high throughput drug discovery via screening of natural product libraries against biological screens. The aim is to characterize biologically active compounds that can serve as precursors for drug development. Given the variety and size of libraries available today, data collection, comparison of results from multiple screenings, information flow and knowledge discovery become critical elements of high throughput screening. Figure 5 gives the high throughput scenario at the drug discovery research center.

There are 3 main research groups in the center: Sample Supply, Bioassay and Chemistry (CNPR, 2001). The Sample Supply group is responsible for the acquisition of diverse samples from natural sources (e.g. plant, marine organisms and fungi), preparation of extracts in a format suitable for high throughput screening and the maintenance of a culture collection of micro-organisms. As for the Bioassay group, the biologists apply the latest developments in molecular and cell biology and in high throughput screening to develop biological assays and run them to differentiate molecules in extracts that have potentially useful pharmaceutical activities. With regard to the chemists in the Chemistry group, they are involved in the isolation, purification and identification of molecules that are responsible for biological activity and that have potential pharmaceutical applications.

The systems of KOOP have been employed in the daily work processes of the Bioassay group to collect and analyze data generated from the high throughput screening tests (Liu, 2000). Although the biologists run different screens concurrently, their working processes are very similar with each other. In general, the processes include: 1) putting plates of sample extracts into the machine to conduct the screening, 2) collecting results generated by the machine, 3) preprocessing data, calculating various attributes of the samples and plates, and importing data into the database, 4) performing data analysis, 5) selecting active sample extracts that exhibit good inhibition level for the next round testing. Upon completion of each screen, useful information are written into the data warehouses so as to facilitate cross screens comparison and reports generation. The KOOP template displayed in the Figure 2 includes a part of the working processes, involving various bubbles like samples ranking, graphics plotting, database query, and message (via email) sending systems. Sharing the same KOOP template, the biologists personalize their own templates with different parameters to satisfy their own process requirements (such as screen name, plates range, inhibition cutoff value).

Before the approach of KOOP, the Bioassay biologists were using an in-house software with similar functionality. The software is a standalone application without workflow nor automation. The biologists have to become familiar with the software so as to perform different data processes via clicking different buttons (on the user interface) manually. However, the scientists have to process a large number of plates. To get the results, the biologists had to spend a long time in front of the computer interacting with the software. Furthermore, most the

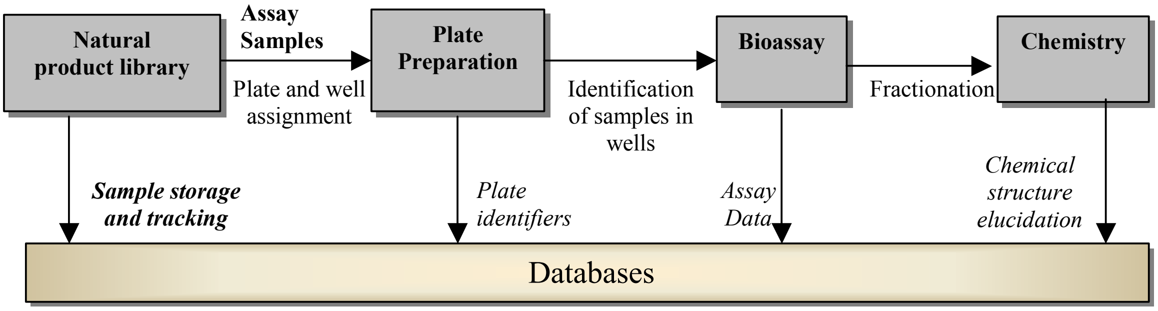

Figure 5: High throughput screening at the drug discovery centre.

biologists perform the data processes during office time. This leads to poor performance due to the increase in usage and the finite processing resources available. With the KOOPlatform, the biologists can set the time schedule to run their processes, for example, arranging server to run certain templates during the night via the scheduler provided. Once the computation is completed, the users can view the results as and when needed.

The other limitation of the old system is the lack of facility to store the parameters that the biologist has set. Once a user quits the software, all parameters set will be lost (except for those have been recorded into the database). These parameters are important as they reflect how the processes are operated and how business decisions are made. The KOOP systems address this by recording down all the parameters and allowing them to be retrieved at any time by anyone who has the access. This is especially useful for newcomers who are still unfamiliar with the business operations.

A useful example is a KOOP template which has been developed for the Bioassay biologists in the screening center. This template is modeled to handle biological process which is very much domain knowledge based. The process involves the selection of extracts that have test results exceeding a certain cutoff, the value of which tends to vary with each screen, experimental conditions, types of samples, etc. In some cases, the biologist has to be personally involved in the determination of the cutoff value, using his domain knowledge and experience. Such implicit knowledge is embedded in various database queries built via a bubble named “BioQuest”. This bubble provides a simple user interface for users who are unfamiliar with SQL. Based on the query results generated, the biologist may decide the further process using an “If-Then” or “True-False” link to other data processing bubbles. For example, if the quantity of the extracts in the query result is insufficient for testing purpose, the biologist may then decide to link another bubble to select all records for next round testing, or a bubble to do further filtering otherwise.

The collection of the above implicit and hidden knowledge is significant for the biologists in improving the quality and the efficiency of the data analysis. However, it is still not easy for them to find, understand and use the knowledge conveniently. The demand of building a meaningful knowledge repository and an artificial intelligent system for the Bioassay group to help choosing active sample extracts is growing. With the information embedded in the KOOPs, data mining would be a good choice to assist the task. The skills of mining classification, association rules and clustering are to be considered to apply to the data collected.

5. FUTURE WORK AND CONCLUSION

KOOP is probably one of the world first hybrid workflow-agent based systems deployed to solve life sciences problems. The strength of KOOP lies in its ability to facilitate data interoperability among distributed legacy systems. This allows real world processes to be modelled end-to-end as software applications.

KOOP is presently being enhanced to allow the inclusion of third party EJBs (Enterprise Java Beans), Java Beans and Java classes as bubbles as well. Eventually, the systems will be extended to absorb non-Java applications via various technologies (such as Java Native Interface, CORBA).

The integration of data mining technologies (based on domain expertise) is another area that is under going further investigation. An implementation which involves various data queries, class classification algorithms and class discovery algorithms have recently been deployed at National Cancer Center to analyse DNA expression data (NCC, 2001).

As the KOOPlatform is lacking in terms of resource management, research are now being carried out to integrate KOOP with various Grid implementation (BioGrid, 2001) like LSF (Platform, 2001) and Globus (Globus, 2002).

KOOP is a relevant new approach for data, applications and knowledge management. With KOOP, users working in the high throughput data collection and analysis area are able to not only integrate, automate, and personalize their business processes, but also maximise the potential of the implicit and hidden knowledge embedded.

REFERENCES

CNPR, 2001. http://www.cnpr.nus.edu.sg.

BioGrid, 2001. http://www.bic.nus.edu.sg/biogrid/.

Globus, 2002. http://www.globus.org/.

Liu, H.Q. 2000. KOOP and High Throughput Screening in Drug Discovery Research. Technical Report, BioInformatics Centre, National University of Singapore.

Lim, T.S. 2000. An Introduction to Knowledge & Object Oriented Programming, Technical Report, KOOPrime Pte Ltd.

Lim, T.S. 2001. KOOP Version 2,.White Paper. KOOPrime Pte Ltd.

Platform, 2001. Building Production Grids with Platform Computing.

NCC, 2001. http://www.omniarray.com/koop.html.