

A Pattern for reactive mobile agent in Genome database annotation

Diego Bonura

Dipartimento di Matematica e
Informatica
Università di Camerino
Email: diego.bonura@unicam.it

Rosario Culmone

Dipartimento di Matematica e
Informatica
Università di Camerino
Email: rosario.culmone@unicam.it

Mauro Angeletti

Dipartimento di Biologia Molecolare
Cellulare Animale
Università di Camerino
Email: mauro.angeletti@unicam.it

Keyword: active database, pattern, agent, genome database

Abstract

The aim of this work is to provide a model to integrate a reactive multi-agent system and active database. The purpose is to allow a quick spreading of information after a database updating. The proposed solution have been applied over BioAgent framework and for the notation context of genome-wide annotation.

Context

We can distinguish an active database from a passive one by its ability of executing automatically some operations when an event occurs. These operations are defined as “trigger” and they have an Event/Condition/Action structure, where an event specifies the reason of the trigger execution (usually the SQL instructions: “insert”, “delete” and “update”), the condition verifies when an action must be executed and finally, the action defines the operations to be executed (1). Generally triggers are used to make dynamic data consistence. We can use this rule to notify internal events, also out of DBMS .

Usually genetic informations too are managed by DBMS and through web applications are made reasoned queries. Often, but not always reasoned queries cover the user total questions, and also users are obliged to do a lot of queries to monitor the information evolutions on different DBMS. The solution here suggested allow to work out, in an automatic way, informations contained in a set of DBMS, moving activities usually carried out by users, in specialized agents. An agent is defined as a program which can move itself through the network, with the aim to execute its tasks (2).

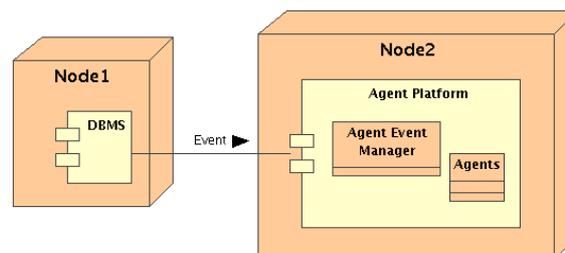


Figure 1

Architecture

Actually the use and fruition of informations contained in genic database needs a constant interaction between user and DBMS front/end. This interaction between the researcher and the informations happens through the fruition of reasoned web pages. These web pages, usually contains forms that have to be compiled and sent to the web application, which once received the request, it returns to the browser, data extracted from DBMS.

But in this way we meet some problems: first of all web pages containing modules for database interrogations, are already predetermined and they will have fields and queries lists fixed by the DBMS and the front/end manager. Therefore it could happen that user finds himself in the condition that he cannot use this service because of the lack of a useful query or form for his purpose. The second and more important problem is that the user is obliged to

constantly monitor the database through continuous sending of requests to web application.

This work fundamental intuition is to join the agent concept (so the code mobility) with the domain of the genetic researches on specialized database. In this way user will have not to interact directly with pre-determined front/end, but to define (through high-level languages) an agent behavior, to fill up it with personalized information and to make it move through the network, towards DBMS satisfying its own interests.

In way to realize the model previously proposed, every operation which modifies the database (such as “insert”, “delete”, “update”), makes start a trigger which sends a message containing modified, inserted or deleted data. The message can be realized in different ways, it depends on the technology used by DBMS triggers (for example through a socket on a specific port). An important feature of the model is that messages are unidirectional (from DBMS towards the management agents system) and that are also datagrams (without confirmation of reception). In this way we have the guarantee that the model proposed doesn't modify operation terminations and data consistence of the database.

The mobile agent platform must have an event manager. The manager of events received from triggers, is able to identify the source and the operation through the event parameters. An agent which arrives in a platform that has this service, says to the event manager what kind of informations it needs. The manager will send to the agent only this informations. Finally the agent will inform the manager when it will not need its events anymore.

The main advantages of this solution are: the high autonomy between the DBMS (which contains the informations) and the agent (created “ad hoc” by the user), so guaranteeing a high independence amongst modules (see Figure 2).

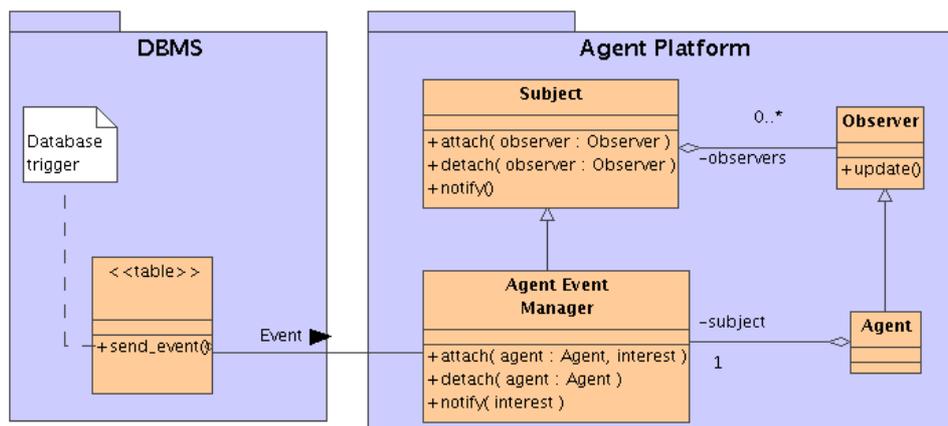


Figure 2

Model instance: Genome annotation

The genome annotation process can be dissected in the following major steps:

- retrieving data and assembling genomes;
- computationally finding genes and other sequence-level features;
- computationally determining homology, function, and other relationships;
- genome-wide structural modeling of gene products;

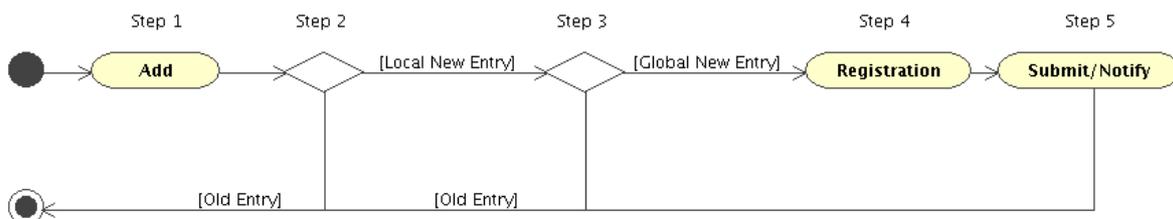


Figure 3

The aim of the project is the application of event-driven agents technology on the first steps of the genome annotation process.

Steps in the annotation process include the following:

- retrieving data and assembling genomes;
- computationally finding genes and other sequence-level features;
- computationally determining homology, function, and other relationships;
- genome-wide structural modeling of gene products;
- analyzing and modeling pathways and systems; and
- managing, accessing, and visualizing data.

Recently Natalia Maltsev (Argonne National Laboratory, ANL) and colleagues described ANL's WIT system (<http://wit.mcs.anl.gov/WIT2>) . WIT was designed and implemented to support genetic sequence and comparative analysis of sequenced genomes and metabolic reconstructions from sequence data. WIT has been designed supposing that the analysis of a large number of phylogenetically diverse genomes can add much to the understanding of higher-level functional subsystems and major physiological designs. In WIT the authors reported a new method for using conserved clusters of genes from numerous genomes to predict functional coupling between genes. Several authors believe the precision of prediction and the amount of accessible functional coupling will increase dramatically as more genomes are added.

So the early steps in the genome annotation process (expecially retrieving data) are critical for the overall success of the annotation analysis.

In this framework, the announcement that a new entry has been submitted to the local database is key event that should be properly "forwarded" to other users (subscribers) which could obtain benefits (for the annotation process) by the informations contained into the new entry.

So, when a new sequence entry is added to the local sequence data base (the event, named step 1 in the flow-chart), the agent will give the announcement to other agents that "something new" had arrived and has to be processed (see step 2 on genome process flow, above).

The event is not merely a data entry event, but it is composed by several on-site tasks, including a pattern matching goal to find out if the sequence entry is really a "new" one (see step 3 in the flow-chart in fig.3).

Other agents will be in charge to guide the registration procedure where a new sequence entry is formally registered as new one into the NCBI database (see step 4).

Finally the sequence entry will be submitted/notified (see step 5) to several organizations who are currently working on gene annotations (adopting the proposed model) (see Genome Annotation Consortium for reference).

The present work provides a specific tool to improve the speed of the genome annotation process.

Finally it should be considered that Extensive annotation increases the value of a genome database remarkably. Furthermore the genome annotation is always to be considered as a "work in progress", because large experimental data sets have to be integrated into the annotation process and added even after finishing a genome project.

In this context, event-triggered genome databases are very promising.

(1) J. Widom and S. Ceri. *Active Database Systems: Triggers and Rules for Advanced Database Processing*. Morgan Kaufmann, San Francisco, California, 1996.

(2) Comuncation of the ACM jurnal, *Intelligen Agent*, Vol. 37, No. 7, July 1994.