TOWARDS A COMPLETE DATA MANAGEMENT FRAMEWORK BASED ON INTELLIGENT AGENTS

Iulian Alexandru Negroiu, Octavian Paul Rotaru, Mircea Petrescu
University "Politehnica" Bucharest Romania

Keywords: Data Management, Data Processing, Agent Based Framework, Intelligent Agent.

Abstract: Applications are more and more complex and the volume of information has an exponentially growth nowadays. In these conditions a huge amount of information needs to be processed while the processing time and power should be kept to a minimum. The increasing amount of data transferred over the Internet and other networks, which are open to a big number of clients, was reflected in the growth of the distributed information systems. Also, there are a multitude of servers distributed among remote locations, which are serving the same purposes. In such cases, traditional models of distributed computing, caching, concurrency control, e.g. are inappropriate in overcoming the actual efficiency problems and in supporting the development of complex applications. We believe that intelligent and autonomous agents can solve this problem. This paper opens the research for creating a complete data management intelligent agent based framework. The possible areas that can be handled using agents are identified and discussed together with the required agents and agencies, while attempting to provide a bird's eye architectural view of the proposed framework.

1 INTRODUCTION

The increasing complexity of the applications and information systems leads to a growth in the need of using distributed models and complex multi-server solutions in the development of computing applications. Such software systems may use both wireless and wired networks, may have time constraints to be met and may also require to access low memory devices.

In these technical conditions there is a need to find a complete approach for managing data in the entire business process of an application. We believe that a framework based on intelligent agents can constitute a suitable solution for the above stated problem. We aim to identify the data management areas that can be handled using agents and the agents and agencies that are required to cover the identified areas.

The framework addresses some of the main performance issues caused by the load of data, distribution of the information systems or network traffic. Other very important aspect of the framework is the flexibility to changes and uses of data information systems that are supported.

Changing of data while offline or online, or by external systems can no longer be a tremendous problem and may require minimum or even no work at all in adapting the system at runtime. Specialized intelligent agents that compose the framework can perform actions like constraint checking, object locking, and queries.

2 THE FRAMEWORK

A list of well-defined intelligent agents that perform different actions having different scope, can work together in order to compose an agents' based framework. The framework itself can be seen as an abstract agent that resides in the system, mainly in the logical layer, but having tentacles in data and presentation layer as well, and that reacts to different changes of the system having as the main goal the complete management of the data and information.

The framework will work the same as an external system that integrates perfectly into the actual environment. The minimum required agents that should be part of the proposed framework, in order to cover the management of data in an application, are:

Alexandru Negroiu I., Paul Rotaru O. and Petrescu M. (2006).
TOWARDS A COMPLETE DATA MANAGEMENT FRAMEWORK BASED ON INTELLIGENT AGENTS.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - AIDSS*, pages 339-342
DOI: 10.5220/0002460103390342

- Constraint checking agents (Constraint Checking Agency in Figure 1);
- Intelligent query agents (Intelligent Query Agency in Figure 1);
- Statistics collect agents (Statistics Collection Agency in Figure 1);
- Statistics process agents (Statistics Processing Agency in Figure 1);
- Caching update agents (Caching Maintenance Agency in Figure 1);
- Caching decision agents (Caching Management Agency in Figure 1);
- Resource Locking Agents (Locking Agency in Figure 1);
- Mediation agents for multi source information access (Mediation Agency in Figure 1);
- Dispatcher, coordination and supervisor agents (Coordination Agency in Figure 1).

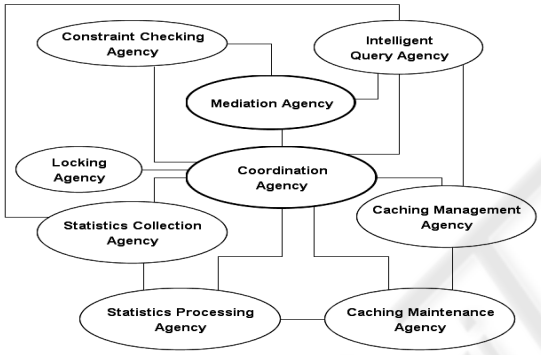

Figure 1: The architecture of the proposed data management agent-based framework.

3 FRAMEWORK'S AGENCIES

All the agencies presented above in Figure 1 will be discussed in this section in order to clearly define their purpose and provided services, justify their presence in the framework and present their interactions with other agents.

3.1 Constraint Checking Agents

Integrity constraints are valuable tools for enforcing consistency of the data in a database. In order to ensure integrity and consistency over the data spanning multiple distributed information systems one has to check the global constraints applied in the specified situation.

A simple modification of data on one site can cause the violation of a global constraint. The check for such violation involves accessing related data from multiple sites. In consequence, one local constraint checker can generate sub-constraint checkers on the remote data by using multiple remote agents. The development of a framework or agent that checks the global integrity constraints of a system is analyzed in (Madiraju, 2004).

Having an agent-based system that implements the constraint checking mechanism will allow the decentralization of the remote checking by having an intelligent decision based solution in a concurrent environment.

An agent based constraint-checking solution for federative database cannot be separated from the transaction processing, and therefore it will require the implementation of the Two Phase Commit (2PC) (Ozsu, 1999) algorithm. 2PC is suitable to be implemented using agents since anyway its logic is distributed between participants, which will execute the local sub-transaction and vote if to commit or rollback, and a coordination process, which will take the global commit or rollback decision based on the votes of the participants. It must be noted that the concept of heterogeneous two phase commit and support for it took nearly a decade to get implemented by DBMS vendors. (Breitbart, 2000) mentions several early efforts in this area.

Another important application of intelligent agents is the verification of constraints using partial information. This is mainly relevant in distributed environments where accessing remote located data is usually very costly. Even in some centralized databases the verification of the constraints using partial information can lead to performance improvements, if the verification is done without using the data stored into the database, and an intelligent agent can also handle it. Information regarding the theory of constraint verification using partial information can be found in (Gupta, 1994).

3.2 Intelligent Query Agents

Another important part of the framework consists in querying the data from the informational systems. The huge volume of information as well as communication bandwidth limitations, time constraints, multiplicity of data formats and maybe the most important, the need for openness and intuitive human interfaces makes the querying layer of the data management system to a crucial component of any information system.

This layer should handle the translation of the requests in queries and sub queries, spread among the data sources in accordance with the data needed, while giving the impression of a single, unified, homogeneous data source.

Researches in the field have come up with agents' models like AQUIRE (Das, 2002) that makes use of combinations of planning and traditional database query optimization techniques and agent-oriented retrieval of data from distributed data sources.

3.3 Statistic Agents

Gathering and processing statistics may become an important part of an agent-based framework like the one we propose. Intelligent query agents, caching agents, resource locking agents etc. all of them need a decision support mechanism in order to complete their own goals with the expected level of performance. In the decision mechanism of an agent, both reactivity to changes in the environment and commitment to long-term courses of actions (behaviours) must be present.

In order to preserve and improve the performance of an agent, gathering and processing statistics become mandatory. Agents are very well suited for such a job, being able to reside in the application and to gather statistics on changes in the environment.

3.4 Caching Agents

A significant portion of an application's resources is consumed by I/O operations, which usually are data accesses. The data access operations are usually the bottlenecks of any software system, especially if they require network transport as well. It is therefore required to implement the data access modules and/or components as efficient as possible. Caching is a technique that can drastically improve the performance of any database application. Due to caching multiple read operations for the same data are avoided. Caching decisions and operations can be outsourced to agents, but these agents will continuously require collaborating and communicating with the intelligent query agents.

In case of a distributed database the caching agents can also be distributed at every node and the data can be also cached in files on the disk and not necessary into the memory. A cached data validity verification agent is also required in order to check if the cached data is still valid or not. "Spying" all the data modification requests handled can do the validity verification. The caching agents will perform all the activities and make all the decisions shown in Figure 2. The caching update agents will handle the activities of putting and getting data from cache, while the caching decision agents will decide if caching is required or not for the retrieved data and if the data contained by the cache is still valid or not.

The type of cache to be used, Primed Cache or Demand Cache (Nock, 2004), is more an architectural decision that a decision that can be taken by an agent. The framework should provide agents able to handle both on demand and primed caching techniques, as well as combined caching strategies, in order to cover the requirements of all kinds of applications.

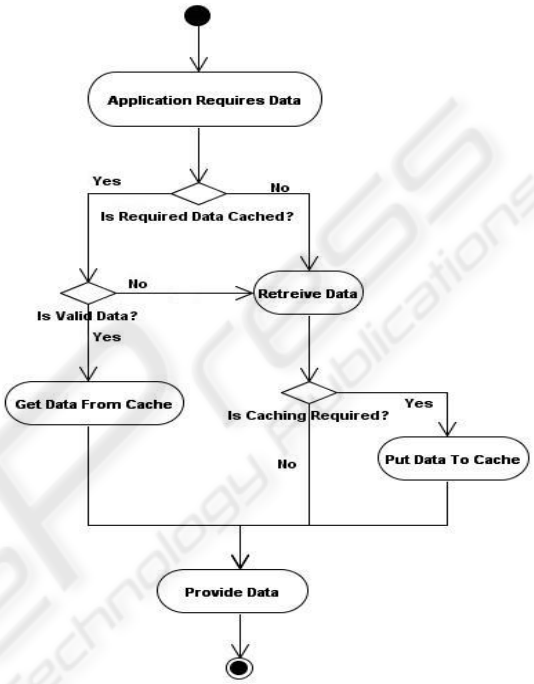

Figure 2: Caching Activity Flow.

3.5 Resource Locking Agents

Multi-user applications needs take care of resource locking in order to avoid the situation in which more than one user attempts to update the same simultaneously. Files, database objects, connections, etc. are all resources that requires to be locked by a user whenever requiring their usage. Intelligent agents can handle both the identification of the critical resources that requires locking and the locking and unlocking mechanisms.

An important problem that needs to be tackled by the framework is the granularity of the locks, which influences heavily the overall performance of the system. Choosing the unit of locking is a trade-off between concurrency and performance. The concurrency is low when the locking unit is big, while the performance can be affected in case the locking unit is too small due to the computational overhead introduced by the big number of locks.

3.6 Mediation Agents

Distributed heterogeneous information systems are built on different technology, may have different access points and may understand different languages. In this conditions there is a need of a fully and easy to customize mechanism that can interact and understand multiple information systems and transport methods. The mediation agents are necessary for maintaining a well-defined and logical communication with the information systems. They can improve performance and can deal with asynchronous communication issues as well.

3.7 Supervisor Agents

The framework we propose contains a big number of agents all of them needing to communicate and all of them having a well defined role in the system. The agents should form a team and communicate one with another in the easiest manner while still maintaining the purpose of creation and giving outputs in each ones specific area.

The goal of maintaining the distribution of roles while still giving the impression of one consistent module can be undertaken by a supervisor and coordinator agent. The existence of such an agent is not a must, the agents of the framework being able to communicate between them. Yet, one supervisor agent will be well noted in the process of customization and completion of the framework with new agents, as well as in the work of integration with the actual application, acting as the project manager of the agents' "team".

We can see the agents of our framework as having roles in a big team. Agents can act and coordinate with other agents depending on their roles. In this idea, the coordination between agents is mainly obtained through common tactics, strategies and observations of actions of team members, rather than explicit communication (Coradeschi, 1998).

4 CONCLUSIONS

Having a better performance and easier to customize systems is an important goal in application development and design. Intelligent agents can offer this two advantages and much more. A data manager framework is an important part of a system, having the role of managing the entire process of data processing and acting as a link between the user logic and the data systems. Such a framework should take decisions regarding the when and what data to be cached, regarding resource locking, how to query and what information system to be queried etc.

This paper presents a bird's eye architectural view of our proposed data management agent based framework. The starting point is the identification of the areas that are possible candidates for agent-based implementation. The paper also discusses all the identified areas in order to justify their appurtenance to the framework and evaluate the consequences.

We believe that an agent-based implementation of such a framework is not only possible but also very useful in regards to performance. Apart from that, the separation of concerns provided by the intelligent agents increases the overall performance of the application. Furthermore, in case of distributed applications collaborative intelligent agents seam to be the logical choice.

REFERENCES

Praveen Madiraju, and Rajshekhar Sunderraman. 2004. *A Mobile Agent Approach for Global Database Constraint Checking*. SAC. Nicosia, Cyprus

Subrata Das, Kurt Shuster, and Curt Wu. 2002. *ACQUIRE: Agent-based Complex Query and Information Retreival Engine*. AAMAS. Bologna, Italy.

M. Tamer Ozsu, Patrick Valduriez. 1999. *Principles of Distributed Database Systems 2nd edition*. Prentice Hall Inc.

Breitbart, Y. 1990. *Multidatabase Interoperability*. Sigmod Record.

Ashish Gupta, Yehoshua Sagiv, Jeffrey D. Ullman, Jennifer Wildom. 1994. *Constraint Checking with Partial Information*. 13th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Minneapolis. pg. 45 - 55.

Clifton Nock. 2004. *Data Access Patterns: Database Interactions in Object-Oriented Applications*. Addison-Wesley, Pearson Education.

Silvia Coradeschi and Lars Karlsson. 1998. *A role-Based Decision-Mechanism for Teams of Reactive and Coordinating Agents*. RoboCup-97: The First Robot World Cup Soccer Games and Conferences. Springer Verlag Lecture Notes in Artificial Intelligence (LNAI)

Stan Franklin and Art Graesser. 1996. *Is it an Agent, or just a program?: A Taxonomy for Autonomous Agents*. 3rd International Workshop on Agent Theories, Architectures and Languages, Springer-Verlag.

Dr. Bruce C. Gabrielson, Mr. Leonid Kunin. 1999. *Enhacing Near Real-time Security Using an Intelligent Agent Approach*. IA-COP Information Assurance Workshop, Advanced Information Technology Joint Program Office, McLean, VA.