A Fault-Tolerant Multi-Agent Framework

Z. Guessoum
LIP6, OASIS team
4 place Jussieu, 75252 cedex 5
(331) 44 27 87 43
Zahia.Guessoum@lip6.fr

J.-P. Briot
LIP6, OASIS team
4 place Jussieu, 75252 cedex 5
(331) 44 27 36 67
Jean-Pierre.Briot@lip6.fr

S. Charpentier
LIP6, OASIS team
4 place Jussieu, 75252 cedex 5
(331) 44 27 87 43

O. Marin
LIP6, SRC team
4 place Jussieu, 75252 cedex 5
(331) 44 27 34 23
Olivier.Marin@lip6.fr

P. Sens
LIP6, SRC team
4 place Jussieu, 75252 cedex 5
(331) 44 27 87 65
Pierre.Sens@lip6.fr

ABSTRACT

To make large-scale multi-agent systems reliable, we propose an adaptive application of replication strategies. Critical agents are replicated to avoid failures. As criticality of agents may evolve during the course of computation and problem solving, we need to dynamically and automatically adapt the number of replicas of agents**,** in order to maximize their reliability and availability based on available resources. We are studying an approach and mechanisms for evaluating the criticality of a given agent and for deciding what strategy to apply (e.g., active replication, passive) and how to parameterize it (e.g., number of replicas).

1. INTRODUCTION

As distributed systems, multi-agent systems (MAS) are exposed to possibility of failure of their hardware and/or software components. The failure of one component can often evolve into the failure of the whole system. To make these large-scale systems reliable, an obvious solution is the introduction of redundancy: duplication (replication) of the critical components.

Replication mechanisms have been successfully applied for various distributed applications, e.g. data-bases. But in most cases, replication is decided by the programmer and applied statically, before the application starts. This works fine because the criticality of components (e.g., main servers) may be well identified and remains stable during the application run. Opposite to that, in the case of dynamic and adaptive multi-agent applications, the criticality of agents may evolve dynamically during the course of computation. Moreover, the available resources are often limited. Thus, simultaneous replication of all the components of a large-scale system is not feasible. The idea is then to automatically and dynamically apply replication mechanisms *where* (to which agents) and *when* it is most needed.

In this paper we will describe our approach to this objective. We first present the replication mechanism and the framework (named DarX) that we developed to replicate agents. We then describe a new model to evaluate dynamically the criticality of agents.

2. DARX

Replication of data and/or computation is an effective way to achieve fault tolerance in distributed systems. A replicated software component is defined as a software component that possesses a representation on two or more hosts.

Several replication strategies (mainly, active and passive) can be used to replicate agents. Each strategy has its pros and cons, the tradeoff being recovery speed versus overhead. Thus, the choice of the most suitable strategy relies on the environment context.

In most multi-agent applications, the environment context is very dynamic. So, the choice of the replication strategy of each component, which relies on a part of this environment, must be determined dynamically and adapted to the environment changes. Moreover, a MAS component which can be very critical at a moment can loose its critically later. If we consider the replication cost which is very high, the number of replicas of these components must be therefore dynamically updated. Thus, the solution we propose allows to dynamically adapt the number of replicas and the replication strategy. This solution is provided by the framework DarX [4].

DarX is a framework to design reliable distributed applications which include a set of distributed communicating agents. Each agent can be replicated an unlimited number of times and with different replication strategies (passive and active). DarX includes group membership management to dynamically add or remove replicas. It also provides atomic and ordered multi-cast for the replication groups' internal communication.

A replication group is an opaque entity underlying every application agent. The number of replicas and the internal strategy of a specific agent are totally hidden to the other application agents. Each replication group has exactly one leader which communicates with the other agents. The leader also checks the liveness of each replica and is responsible for reliable broadcasting. In case of failure of a leader, a new one is automatically elected among the set of remaining replicas.

DarX provides the needed adaptive mechanisms to replicate agents and to modify the replication strategy. Meanwhile, we cannot always replicate all the agents of the system because the available resources are usually limited. The problem therefore is to determine the most critical agents and then the needed number of replicas of these agents.

We distinguish two cases: 1) the agent's criticality is static and 2) the agent's criticality is dynamic. In the first case, critical agents can be identified by the designer and can be replicated by the programmer before run time. In the second case, MASs may have dynamic organization structures. So, the agents criticality cannot be determined before run time. The agent criticality can be therefore based on these dynamic organizational structures. The problem is how to determine dynamically these structures to evaluate the agent criticality?

Thus, we propose a new approach for observing the domain agents and evaluating dynamically their criticality. This approach is based on two kinds of information: semantic-level information and system-level information.

3. ADAPTIVE CONTROL OF REPLICATION

We will now detail our approach for dynamically evaluating criticality of each agent in order to perform dynamic replication where and when best needed.

We want some automatic mechanism for generality reasons. Our approach is not related to a specific interaction language or application domain. Also agents can be either reactive or cognitive. We just suppose that they communicate with some agent communication language [1]. But in order to be efficient, we also need some prior input from the designer of the application. This designer can choose among several approaches of replication: static and dynamic.

In the proposed dynamic approach, the agent criticality relies on two kinds of information:

- System-level information. It will be based on standard measurements (communication load, processing time...). We are currently evaluating their significance to measure the activity of an agent.

- Semantic-level information. Several aspects may be considered (importance of agents, dependence of agents, importance of messages ...). We decided to use the concept of role, because it captures the importance of an agent in an organization, and its dependencies to other agents.

The application designer will manually valuate criticality of the roles, corresponding to their "importance" in the organization and in the computation. In order to track the dynamical adoption of roles by agents, we propose a role recognition method. This method is based on the observation of the agent execution and their interactions to recognize the roles of each agent and to evaluate his processing activity. This is used to dynamically compute the criticality of an agent.

In order to collect the data, we associate an observation module to each DarxServer on each machine. It will collect events (provided by DarxServer). We consider two cases. In the first case, each agent displays explicitly his roles or interaction protocols. The roles of each agent are thus easily deduced from interaction events.

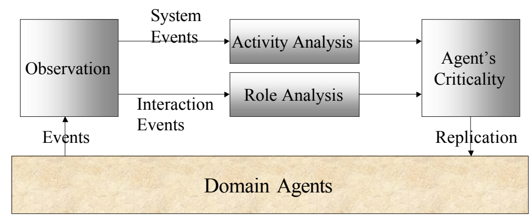

Fig. 1: General architecture for replication control

In the second case, agents do not display their roles nor their interaction protocols. The agent roles are deduced from the interaction events by the *role analysis* module. In this analysis, attention is focused on the precise ordering of interaction events (exchanged messages). The *role module* captures and represents the sequence of interaction events resulting from the domain agent interactions (sent and received messages). These interaction events are then used to determine the roles of the agents.

To represent the agent interactions, several methods have been proposed such as state machines and Petri nets. For our application, state machines provide a well suitable representation. Each role interaction model is represented by a machine state. A library of role definitions is used to recognize the active roles.

4. CONCLUSION

Large-scale MASs are often distributed and must run without any interruption. To make these systems reliable, we proposed a new approach to evaluate dynamically the criticality of agents [3]. This approach is based on the concepts of roles and degree of activity. The agent criticality is then used to replicate agents in order to maximize their reliability and availability based on available resources.

To validate the proposed approach, we realized a fault-tolerant framework (DarX). The integration of DarX with a multi-agent platform, such as DIMA [2], provides a generic fault-tolerant multi-agent platform. In order to validate this platform, two small applications have been developed (meetings scheduling and crisis management system). They are intended at evaluating our model and architecture viability. The obtained results are interesting and promising. However, more experiments with real-life applications are needed to validate the proposed approach.

5. REFERENCES

[1] FIPA 97. Specification. Part 2, *Agent Communication Language*, www.fipa.org.

[2] Z. Guessoum and J.-P. Briot. From active objects to autonomous agents. In Special Series on Actors and Agents, edited by D. Kafura and J.-P. Briot, IEEE Concurrency, 7(3):68-76, July-September 1999.

[3] J.-P. Briot, Z. Guessoum, S. Chapentier, S. Aknine, O. Marin and P. Sens. Dynamic Adaptation of Replication Strategies for Reliable Agents. In Adaptive Agents and Multi-Agent Systems, pp. 10-19, AISB02. Mar 2002

[4] O. Marin, P. Sens, J.-P. Briot and Z. Guessoum, Towards Adaptive Fault-Tolerance For Distributed Multi-agent

Systems, in *Proceedings of ERSADS'2001*, Bertinoro, Italy, pp. 195-201, May 2001.