AN ADAPTIVE AND DEPENDABLE SYSTEM CONSIDERING INTERDEPENDENCE BETWEEN AGENTS

Keinosuke Matsumoto, Tomoaki Maruo, Akifumi Tanimoto and Naoki Mori
Graduate School of Engineering, Osaka Prefecture University, 1-1 Gakuen-cho, Nakaku, Sakai, Osaka, Japan

Keywords: Dependability, Interdependence graph, Monitoring, Replication, Intelligent systems.

Abstract: A multiagent system (MAS) has recently gained public attention as a method to solve competition and cooperation in distributed systems. However, MAS's vulnerability due to the propagation of failure prevents it from being applied to a large-scale system. This paper proposes an approach to improve the efficiency and reliability of distributed systems. The approach monitors messages between agents to detect undesirable behaviours such as failures. Collecting the information, the system generates global information of interdependence between agents and expresses it in a graph. This interdependence graph enables us to detect or predict undesirable behaviours. This paper also shows that the system can optimize performance of a MAS and improve adaptively its reliability under complicated and dynamic environment by applying the global information acquired from the interdependence graph to a replication system.

1 INTRODUCTION

A multiagent system (MAS) (Weiss, 1999) has recently gained public attention as a method to solve competition and cooperation in distributed systems. However, MAS's vulnerability due to the propagation of failures prevents it from being applied to a large-scale system. It is indispensable for constructing a ubiquitous network society to raise MAS's reliability on a large scale.

This paper proposes an approach to improve the efficiency and reliability of distributed systems. The approach monitors messages between agents to detect undesirable behaviours such as failures. Collecting and reserving the information, it generates global information of interdependence between agents and expresses it in a graph. This interdependence graph enables us to detect or predict undesirable behaviours.

Replication systems (Guerraoui, 1997) may arrange multiple replicas with the same contents on a network. They are regarded as an effective technique for raising the fault tolerance of distributed systems. Such systems have replicas replaced with faulty agents in order to realize fault tolerant applications. However, they are not suitable for large-scale systems because replication cost may increase in proportion to the number of replicas. This paper also proposes an adaptive replication system and replication policies that can efficiently reduce replication cost and raise performance.

2 PROBLEMS OF PREVIOUS MONITORING METHODS

Monitoring is usually adopted as a technique for raising reliability of MAS that includes undesirable behaviours. Some techniques are proposed and problems of these techniques are shown below: An immunity network (Ishida, 1996) depends on knowledge agents have locally. It is a self-diagnostic system, and it becomes very complicated because fault tolerance and performance may be dependent on network configuration. Moreover, a method of Kaminka et al. (Kaminka, 2002) is based on a recognition model of procedure to identify disagreement of states. As the procedure is static and premised on closed systems, this approach cannot be adapted for changing environments. Finally, a method of Horling et al. (Horling, 2001) is a diagnostic system of distributed systems that uses certain fixed failure models. It can optimize system's performance locally, but not globally. As a result, this approach is not suitable for a MAS under open environments where agents are added or deleted on large scale and in real time.

Matsumoto K., Maruo T., Tanimoto A. and Mori N. (2008).
AN ADAPTIVE AND DEPENDABLE SYSTEM CONSIDERING INTERDEPENDENCE BETWEEN AGENTS.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - SAIC*, pages 199-202
DOI: 10.5220/0001669901990202

3 MONITORING METHOD USING INTERDEPENDENCE GRAPH

3.1 Interdependence Graph

Collecting related information, the monitoring system generates global information of interdependence between agents and expresses it in a graph. A domain agent is related to a node as shown in Figure 1. This situation is expressed as labeled graph (N, L, W). It is called an interdependence graph. Where, N is a node set, $L_{i,j}$ is a link from node N_i to N_j, and $W_{i,j}$ is a weight labeled to a link $L_{i,j}$. $W_{i,j}$ corresponds to importance of the interdependence between agent i and j, and n is the total number of nodes. If domain agents are added or deleted, the graph is updated dynamically.

$$N = \{Ni\}\ i=1,\ldots,n \tag{1}$$

$$L = \{Li,j\}\ i=1,\ldots,n,\ j=1,\ldots,n \tag{2}$$

$$W = \{Wi,j\}\ i=1,\ldots,n,\ j=1,\ldots,n \tag{3}$$

3.2 Monitoring Architecture

Monitoring task consists of two processes: acquisition of the information for updating an interdependence graph, and the graph analysis for controlling domain agents. This information is standard indicators like communication load and processing time, agents' characteristics and so on.

The architecture is shown in Figure 2. This distributed observation mechanism corresponds to the organization of agents that react in adaptation to changing environments. There are two kinds of agents: monitoring agents and a host monitoring agent. A monitoring agent is associated with each domain agent, and with the host monitoring agent. The monitoring agent collects the individual information as a monitor. Each monitoring agent

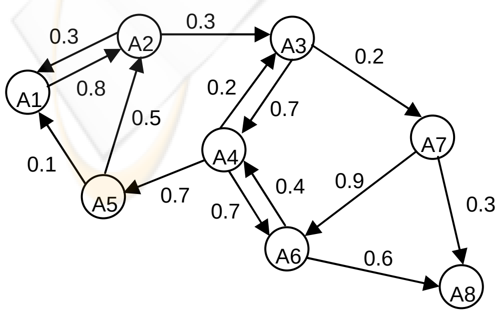

Figure 1: Interdependence graph.

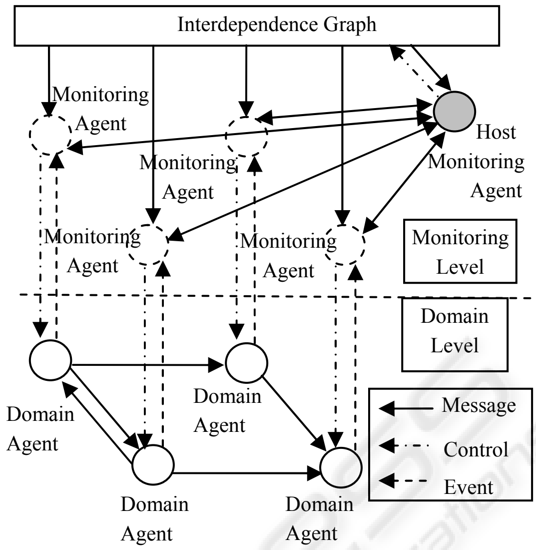

Figure 2: Monitoring architecture.

communicates with the host monitor agent, and transmits local information acquired by monitoring. The host monitor agent makes it into global information such as the total number of exchanged information. The monitoring agent reflects various changes of the domain agent in the interdependence graph. Because agent environment changes in real time, this graph is not static, and it is updated dynamically if agents in the domain level are added or deleted.

3.3 Updating Algorithm of Weights

Indexes for updating algorithm of weights are shown below:

- monitoring time interval Δt
- communication load $Q(\Delta t)$

$$Q(\Delta t) = \text{operator1}(Q_{1,1}(\Delta t), \ldots, Qn,n(\Delta t)) \tag{4}$$

- the number of transmitting messages $NM(\Delta t)$

$$NM(\Delta t)=\text{operator2}(NM_{1,1}(\Delta t), \ldots, NMn,n(\Delta t)) \tag{5}$$

Where $Q_{i,j}(\Delta t)$ and $NM_{i,j}(\Delta t)$ express respectively the amount of communication data and the number of transmitting messages from agent i to j in time interval Δt, operator1 and operator2 are set operators and they are usually average operators.

The outline of the updating algorithm that updates the weights $W_{i,j}$ of the interdependence graph is shown below.

1. Repeat the following for agent j ($j \neq i$).

2. Calculate formulas (6) and (7)

$$Qtemp= (Q_{i,j}(\Delta t)-Q(\Delta t))/Q(\Delta t) \quad (6)$$

$$NMtemp= (NM_{i,j}(\Delta t)-NM(\Delta t))/NM(\Delta t) \quad (7)$$

3. Updating weight $W_{i,j}$

$$W_{i,j}(t+\Delta t)=W_{i,j}(t)+\alpha\times Operator3(Qtemp,NMtemp) \quad (8)$$

4. End of repetition

Parameter α is a discount rate in which a new situation is reflected in the existing weight.

4 ADAPTIVE MULTIAGENT ARCHITECTURE

4.1 Adaptive Replication System

Our adaptive multiagent architecture is shown in Figure 3. This architecture consists of a monitoring system and a replication server that manage domain agents and replicas. In our adaptive replication system, you can adjust dynamically the number of replicas according to each agent's importance obtained from the interdependence graph.

The replication system creates some replication groups, and each group consists of one domain agent and some replicas. The algorithm adjusts adaptively the number of replicas in proportion to the importance of a node.

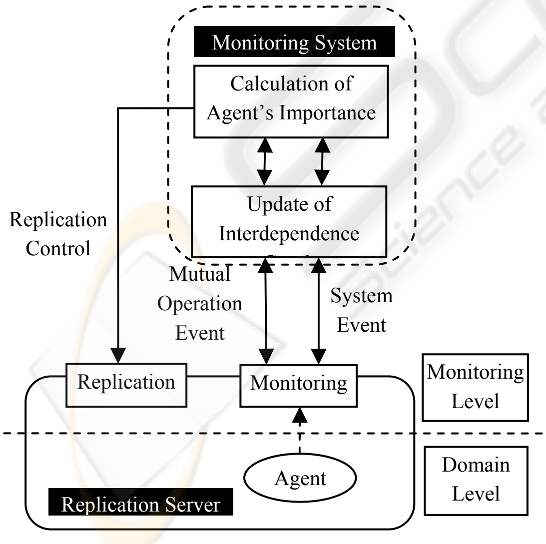

Figure 3: Adaptive multiagent architecture.

4.2 An Algorithm to get the Number of Replicas using Interdependence

The interdependence graph is useful for grasping the influence of failures and the fault tolerance of multiagent systems. The proposed monitoring system estimates an agent's importance by executing a set operation (operator4) on the weights $W_{i,j}$ that are labeled to each agent's input-and-output links. It is shown in formula (9).

$$w_i = operator4\ (W_{i,j}\ j=1, ---, m) \quad (9)$$

w_i is the importance of agent i and m is the degree of node i. The w_i is used to compute the number of replicas (rep_i) for agent i in our adaptive replication system. Formula (10) shows this relation.

$$rep_i = round\ [r_0 + r_{max}\times w_i /W] \quad (10)$$

r_0 is an initial number of replicas, r_{max} is the maximum value of the total number of replicas a system designer sets up beforehand, and W is the sum of all agents' w_i.

5 SIMULATION EXPERIMENT

5.1 Virtual eMarket Multiagent System

The simulation is carried out for a virtual eMarket multiagent system (eMarket MAS):

1. Intermediate-goods producers sell intermediate goods to consumption-goods producers.
2. Consumption-goods producers process the intermediate goods and make consumption goods.
3. Consumers purchase the consumption goods from consumption-goods producers.

A model of the two stage market used in the simulation is shown in Figure 4. In this model, there are two markets:

– Market A: Intermediate-goods market
– Market B: Consumption-goods market

Furthermore, agents constituting markets are classified into the following four kinds of agents according to their roles: Sellers, buyers, buyer/sellers, and market management agents. Sellers and buyers only sell intermediate goods or purchase consumption goods respectively. The buyer/sellers play different roles in each market. They buy intermediate goods in Market A, and sell consumption-goods in Market B. Finally, one

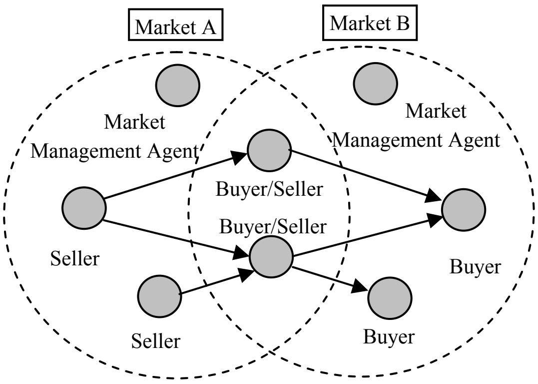

Figure 4: Two stage market model.

market management agent exists in each market, and it controls registration or deletion of agents constituting markets. The market management agent mediates dealings. The markets deal with dealings by auctions, and the scenario is described below:

1. A seller gives a market management agent a reserve price.
2. The market management agent accepts bids from buyers during a certain fixed time.
3. The market management agent cuts a deal with the buyer that presented the highest bid price exceeding the reserve price in a bid deadline.
4. If there is no corresponding bid, dealings are abortive and the seller again presents the price that is less than the previous one.

5.2 Experiment

Fault generators put a total of 100 agents to stop within 10-minute simulations. If an agent exhausts replicas prepared beforehand, the dealing scenario cannot be completed. This means the simulation is a failure. We count the number of successful simulations when we change the total number of replicas r_{max} from 0 to 30. Simulations are performed 40 times in total. Parameters used for the experiment are as follows: Market management agents: 2, buyer/sellers: 50, buyers: 24, sellers: 24, and total 100 agents. Monitoring interval is 500ms. Discount rate α is 1.0. Initial number of replicas r_0 is 1.

The experiment results are shown in Figure 5. When r_{max} is set up as 10, the rate of success reaches 80%. If r_{max} is set as 20 and over, the success rate becomes 100%. It proves that reliability of the system is maintainable by setting r_{max} as 20 and over in this experiment environment, and it also means that replication cost can be held down efficiently. The parameter r_{max} is important for systems and must be set up suitably.

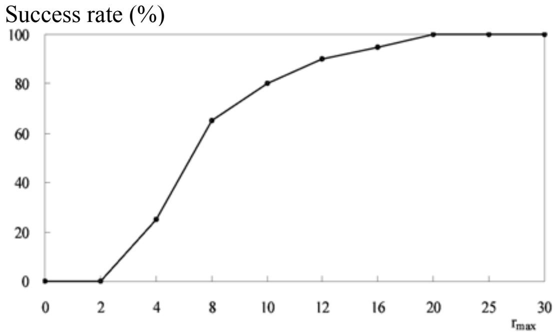

Figure 5: Relationship between r_{max} and success rate.

6 CONCLUSIONS

To improve the efficiency and reliability of multiagent systems, this paper has proposed an algorithm
(i) to update the interdependence graph and
(ii) to adjust adaptively the number of replicas in proportion to the importance of a node using the interdependence graph.
It also has proposed an adaptive replication system that uses global information acquired by monitoring to improve fault tolerance of multiagent systems.

The method has been applied to an e-market MAS. The simulation results show that the method can optimize performance of a MAS and improve its reliability by applying the global information acquired from the interdependence graph to replication systems.

REFERENCES

Weiss, G., 1999. *Multiagent Systems -A Modern Approach to Distributed Artificial Intelligence-*. The MIT Press, pp.79-120.

Guerraoui, R., Schiper, A., 1997. Software-based replication for fault tolerance. IEEE Computer , Vol.30, No.4, pp.68-74.

Ishida, Y., 1996. An Immune Network Approach to Sensor-based Diagnosis by Self-organization. Complex Systems, Vol.10, No.1, pp.73-90.

Kaminka, G. A., Pynadath, D. V., Tambe, M, 2002. Monitoring Teams by Overhearing: A Multi-agent Plan-Recognition Approach. Intelligence Artificial Research, Vol.17, pp.83-135.

Horling, B., Benyo, B., Lesser,V., 2001. Using Self-Diagnosis to Adapt Organizational Structures. Proc. of 5th International Conference on Autonomous Agents, pp.529-536.