

Document downloaded from:

<http://hdl.handle.net/10251/69192>

This paper must be cited as:

Poza-Lujan, J.; Posadas-Yagüe, J.; Simó Ten, JE. (2010). Quality of service and quality of control based protocol to distribute agents. En Distributed Computing and Artificial Intelligence: 7th International Symposium. Springer. 73-80. doi:10.1007/978-3-642-14883-5_10.



The final publication is available at

http://link.springer.com/chapter/10.1007/978-3-642-14883-5_10

Copyright Springer

Additional Information

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-14883-5_10

Quality of Service and Quality of Control Based Protocol to Distribute Agents

Jose-Luis Poza-Luján, Juan-Luis Posadas-Yagüe, José-Enrique Simó-Ten

University Institute of Control Systems and Industrial Computing (ai2). Universidad Politécnica de Valencia. Camino de vera, s/n. 46022 Valencia (Spain)
{jopolu, jposadas, jsimo}@ai2.upv.es

Abstract This paper describes an agent's movement protocol. Additionally, a distributed architecture to implement such protocol is presented. The architecture allows the agents to move in accordance with their requirements. The protocol is based on division and fusion of the agents in their basic components called Logical Sensors. The movement of the agents is based on the quality of services (QoS) and quality of control (QoC) parameters that the system can provide. The protocol is used to know the impact that the movement of the agents may have on the system and obtain the equilibrium points where the impact is minimal.

1. Introduction

Distributed control architectures based on multi-agent systems require a middleware to hide the complexity of the communications to the agents. The middleware can provide the data about QoS to the system, and system can provide this information to the agents. On the other hand, when an agent needs to move between the control nodes, the arrival of the agent have an impact on the global performance of the control node. To provide the support to agents an architecture, called Frame-Sensor-Adapter with Control support (FSACtrl), has been developed. This architecture is based on the model FSA, widely tested on mobile robot, and home automation systems [1]. The middleware of the architecture is based on the Data Distribution Service (DDS) model proposed by The Object Management Group (OMG) [2].

When an agent arrives to a node, competes with the other agents for the resources. The effect of the new agent on the node performance is determined by the Liebig's law of the minimum [3]: similar species competing for the same resources minimizing the availability of the global resources. To minimize the impact of the agent movement the architecture can move and evaluate only fragments of an agent. How the architecture can provide the necessary information to know the impact of the agent movement is the aim of this article.

The article has been organized as follows: Next section introduces the theoretical Concepts that are the base of the architecture on which the protocol shown is developed. The third section describes the components of the architecture and the quality-based supply and demand cycle. The fourth section shows the phases of the protocol. Finally, presents concluding remarks.

2. Middleware, quality of service and quality of control

There are different paradigms of communication with support to quality of service, among them publish-subscribe model is one of the most suitable [4]. DDS provides a platform independent model that is aimed to real-time distributed systems. DDS is based on publish-subscribe communications paradigm. Publish-subscribe components connect information producers (publishers) and consumers (subscribers) and isolate the publishers and the subscribers in time, space and message flow [5].

When a producer (component, agent or application) wants to publish some information, should write it in a Topic by means of a component called Data Writer which is managed by another component called Publisher. Both components, Data Writer and Publisher, are included in another component called Domain Participant. On the other hand, a Topic can delivery messages to both components: Data Readers and Listeners by means of a Subscriber. Data Reader provides the messages when the application requires. Instead of a Listener sends the messages without waiting for the application.-

Quality of Service is defined as the collective effect of service performance, which determines the degree of satisfaction of a user of the service [6]. FIPA defines a set of QoS parameters mainly based on temporal issues of the messages [7]. DDS specification proposes 22 different QoS policies that cover all aspects of communications management: message temporal aspects, data flow and metadata. For example, by means the “Deadline” policy, that determines the maximum time for the message arrival, and the “Time-Based-Filter” policy, that determines the minimum time between two messages, a component can establish a temporal window to receive messages from other components.

In the same way that to evaluate the efficiency of communications uses QoS parameters; control must provide the corresponding parameters, known as quality control parameters. It's considered a perfect control when the signal is sent to the actuator causes the signal measured by the sensor is identical to a reference signal, therefore there is no error between the measured signal and reference signal. Control error is used to modify the signal sent to actuator. The most commonly used QoS parameters are the value of the Integral Absolute value of Error (IAE) and the Integral of the Time and the Absolute value of Error (ITAE). Both parameters allow the system to know how evolve the error, and predict the new action. [8].

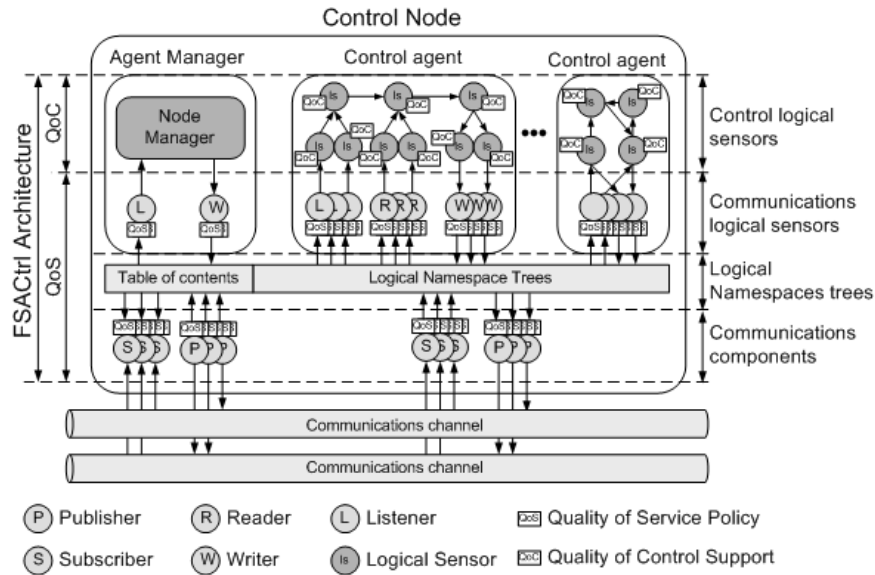


Fig. 1. Components of FSACtrl architecture

3. Distributed architecture

3.1 Components

Figure 1 shows the main components of the architecture FSACtrl [9]. Each control node has a manager agent, the necessary control agents, the communications components to provide support at control agents, and a set of topics to connect the control agents with the communications components, Topics are organized in an ontology called Logical Namespace Tree.

Each control node contains a special ontology called Table of Contents (TOC). TOC describes the control node to the other control nodes of the system. The communications components are the components proposed by the DDS standard. Publishers and Subscribers are the control node communications components and they are accessible to all control agents, whereas Data Writers, Data Readers and Listeners are the control agent communications components and they are exclusive to each control agent. Control algorithms are implemented by the components called Control Logical Sensors that provides the QoC parameters.

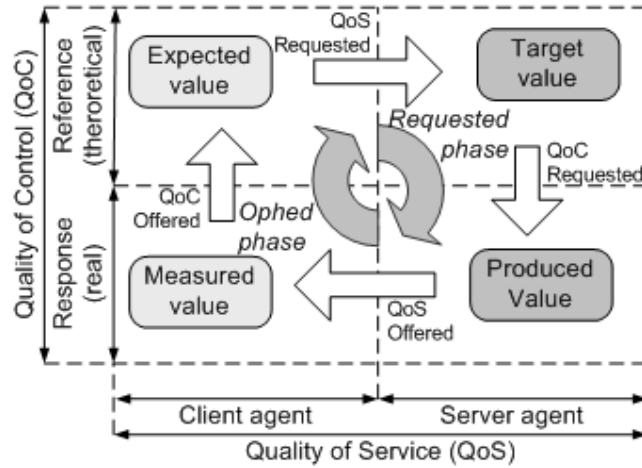


Fig. 2. Both qualities, quality of service and quality of control determine the quality-based supply and demand cycle

Manager agent provides support to all functions of the MAS: processes the request of the control agents, both from within and outside the control node, manage the ontology to connect successfully communications components and control components and mediates in the negotiation based on the QoS and QoC parameters. System manager and LNT are similar to other MAS system as JADE [10].

3.2 QoS and QoC based supply and demand cycle

Figure 2, shows the location of the parameters in the control loop based on a client-server model. Based on the QoS cycle [11], the Quality-based Supply and Demand Cycle adds the QoC. The QoS parameters are associated with the client-server communication and the QoC parameters characterize the relationship between the theoretical expectations and the real results of the control actions, both the client and the server side.

DDS offers a protocol for negotiating the QoS parameters by means the QoS policies. However, when the QoC is included in the negotiation process, it is necessary that the control agent can provide the appropriate protocol and parameters to measure the optimization level.

As seen in figure 2, the QoS and QoC have a fist service request phase to communicate the value needed of the parameters. The second phase, communicates the values offered. Consequently the cycle consists in two phases: the QoS and QoC request, and the QoS and QoC offer. This cycle is known as “quality-based Supply and Demand Cycle”.

4. Moving or cloning agents in the distributed system

4.1 Search destiny control node phase

FSACtrl architecture allows to the agents to perform the same actions than the rest of the MAS [12]. Move or clone an agent involves a set of operations like insert or remove agents. In FSACtrl, insert an agent in a node is done inserting the agent logical sensors. The QoS and the QoC of an agent is easy to calculate because it depends from its internal components. However, when an agent needs change the control node, it's difficult to know the effect of the change in the agent's behaviour.

To know the effect of the node change, the architectures provides the protocol based on the supply and demand cycle. The protocol has two phases. First phase is the selection of the destination node and second phase is the progressive transference of components between the source node and the destination node.

Destination node can be selected based on a set of aspects. To evaluate the adequacy of the destination node FSACtrl uses two aspects: if exists in the destination node the set of logical data that the agent need and if the destination node can conform the QoS and the QoC parameters required by the agent. The logical data available in the destination node is more relevant to select it. The more data matching between the two nodes have, the more similar they are, and consequently probably the agent will work efficiently.

To insert an agent in a control node, the manager agent must check if the destination node can supply all logical data. The logical data that the destination node can't supply are created and associated directly with the corresponding Publisher or Subscriber. Next agent asks to the Node Manager about the QoS ranges. If some agent QoS parameter requirement is out of range, the destination node is discarded.

Node Manager obtains the QoS range values from the LNT. In the LNT every logical data joins a Publisher or a Subscriber with a Data Reader, Data Writer or Listener, these connections has been negotiated previously among the agents and the control node. As the result of this negotiation, the control node can determinate the maximum or minimum values of the QoS parameters.

Usually, if a control node has all logical data needed for the new agent, the QoS of the other agents was only hardly affected because the messages are the same for the new agent. The QoS parameters are directly related with the message type, and only in a few cases, an agent requires a different QoS for the same type of information. However, the Liebig's law of the minimum affects the services offered by the other agents [13], and the QoC of every agent is affected. The aim of the agent movement protocol is to achieve the balance between the QoC and the QoS of the agents. This aspect is described in the next section.

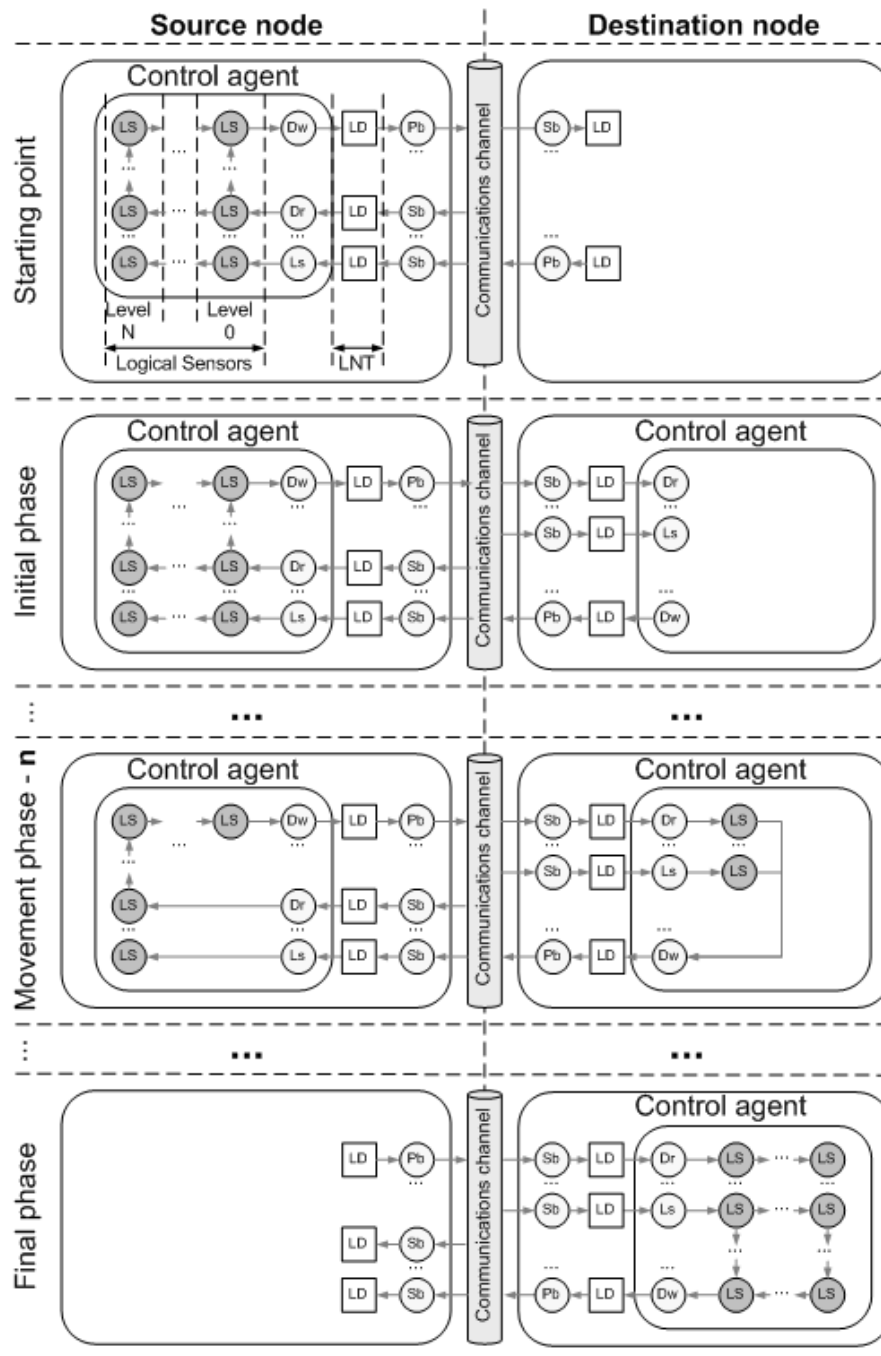


Fig. 3. All phases of the process of an agent movement

4.2 Move or clone control agent phase

The second phase is the logical sensors transference from the source node to the destination node (figure 3). The information about the control agent composition, as the logical data needed, control algorithms or QoS and QoC parameters is contained in the agent specification. The destination node has the Publishers and the Subscribers needed for the communications channel used, if these components are not present is because the destination node can't use this channel and the agent can't be moved. First thing to do is create the destination agent, in the case of the agent need be divided between the two nodes, or use an existing agent, in the case of a merge operation. Then the source Manager selects a Logical Sensor in the source agent and asks the destination Manager to insert the Logical Sensor in the destination node. When the destination agent is inserted, both destination Manager and source Manager creates the same Logical Data to communicate with the Logical Sensors of the source agent, finally launch the new Logical Sensor and stops the old Logical Sensor. The key of the protocol is the running of the Quality-based supply and demand cycle to every Logical Sensor movement phase. Once a cycle has finished, Manager knows the new QoS and QoC values of all the agents in the destination node, and can decide to move other Logical Sensor or undo the last action.

The Logical Sensors that has not been translated send (or receive) the messages to (or from) the moved agents through the LNT. The Manager Agent creates the Data Writer, Data Reader or Listener and the Logical Data needed to supply the old direct connection between the Logical Sensors [14]. Logical Sensors are being moved one to one between the source and the destination node. For every Logical Sensor inserted the QoS and the QoC of the destination node changes, generally global quality decreases. The evolution of the QoS and QoC parameters allows node and agents to regulate the balance between the QoS and the QoC. The agent moved, can decide stop the process at the moment when the QoS and QoC parameters are optimal. The destination node can stop the process if the last Logical Sensor exceeds the QoS and QoC limits determined by the other agents.

5. Conclusions

This article has presented a protocol to move agents between nodes along a distributed system based on an architecture called FSACtrl. The main contribution is the joint use of QoS and QoC parameters to determine, gradually, the impact that an agent has in a control node during the process of movement.

The protocol is used to balance the agent components, usually different control algorithms, between the control nodes. From the location of the agents in the control node, it is possible to determine the optimal composition of the agents to a specific environment.

Currently the formulas to measure the impact in the node and in the agent, in terms of QoS and QoC are being developed. One of the aims of the formulas is provide global the Quality of Node (QoN) and the Quality of Agent (QoA) can be determined by means the combination of the QoS and QoC parameters. With the QoA, the continuous process of divisions and merges of Logical Sensors can stop in the instant that QoS and QoC of all agents are optimal.

Acknowledgments The architecture described in this article is a part of the coordinated project SIDIRELI: Distributed Systems with Limited Resources. Control Kernel and Coordination. Education and Science Department, Spanish Government. CICYT: MICINN: DPI2008-06737-C02-01/02.

References

1. Posadas, J.L., Poza J.L., Simó J.E., Benet G., Blanes, F.: Agent Based Distributed Architecture for Mobile Robot Control. Engineering Applications of Artificial Intelligence. Pergamon Press Ltd. Vol.: 21 N. 6. p.p. 805-823 (2008)
2. Object Management Group (OMG): Data Distribution Service for Real-Time Systems, v1.1. Document formal / 2005-12-04. (2005)
3. Odum, E.P. Fundamentals of Ecology 3rd Edition, W.B. Saunders Company, Philadelphia, (1971)
4. Aurrecochea, C., Campbell, A.T. and L. Hauw.: A Survey of QoS Architectures. ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, Vol. 6 No. 3, pg. 138-151 (1998)
5. Pardo-Castellote, G. OMG Data-Distribution Service: architectural overview. Proceedings of 23rd International Conference on Distributed Computing Systems Workshops. Providence, USA. Vol. 19-22, pp. 200-206 (2003)
6. International Telecommunication Union (ITU). Terms and Definitions Related to Quality of Service and Network Performance Including Dependability. ITU-T Recommendation E.800 (0894) (1994)
7. Foundation for Intelligent Physical Agents. FIPA Quality of Service Ontology Specification, Experimental Doc: XC00094. (2002)
8. R.C. Dorf and R.H. Bishop, Modern Control Systems, 11th Edition, Prentice Hall (2008)
9. Poza, J.L., Posadas, J.L., Simó, J.E.: Middleware with QoS Support to Control Intelligent Systems. 2th International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP, pp.211-216 (2008)
10. Bellifemine F.; Poggi A.; Rimassa G. (1999). Jade: A FIPA-compliant agent framework. Proceedings of PAAM-1999, pp. 97-108.
11. Poza, J.L., Posadas, J.L., Simó, J.E.: From the Queue to the Quality of Service Policy: A Middleware Implementation. In: S. Omatu et al. (Eds.): IWANN 2009, Part II, LNCS 5518, pp. 432-437 (2009)
12. Foundation for Intelligent Physical Agents. FIPA Agent Management Specification, Doc: FIPA00023. (2000)
13. Buhwan Jeong, Hyunbo Cho, Boonserm Kulvatunyou and Albert Jones, "A Multi-Criteria Web Services Composition Problem", In Proceedings of the IEEE International Conference on Information Reuse and Integration, 2007 (IRI 2007), pp. 379-384, IEEE 2007
14. Poza, J.L. Posadas, J.L., Simó, J.E., Benet, G. Distributed Agent Specification for an Intelligent control Architecture. 6th International Workshop on Practical Applications of Agents and Multiagent Systems. IWPAAMS. ISBN 978-84-611-8858-1. (2007)