Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp33-39)

Customized Design and Development of Distributed Control System for Process Control

QURBAN A. MEMON**, and HABIB-UR REHMAN*
EE Department, UAE University, United Arab Emirates

Abstract: In this paper, we discuss customized design and development of Distributed Control System (DCS) for process control using two sub-processes: one at the local level of the network using decentralized agents, and the other for the design of such agents at the global level in the form of operating system to address overall reconfigurability in the presence of real time requirements. The main focus is on the process control in an environment of intelligent field devices and Programmable Logic Controllers (PLC). The three key components namely system architecture, configuration, network and communication parameters are presented in detail. The problems and challenges faced by such a customized design are discussed and remedies presented.

Key-words— Distributed control, Network control system, Profibus, Intelligent agents

1 Background

The need for modularity, integrated diagnostic, decentralization of control, expanding physical setups, and functionality resulted into today's DCS [1]-[15] which is most widely being used for many applications. Presently, the DCS concept is not only applied to the process control and manufacturing automation but it has covered many other areas like power system engineering [1]-[3], nuclear engineering [4], ship engineering [5], and even a lighting system design [6]. The innovative ideas are being applied in this field, for example [6] has made use of the power line as the communication medium for the DCS of the lighting system and is based on a custom built large scale integrated Neuron chip which incorporates both the control and communication features in a three microprocessors based system. However, the focus of this work is the DCS designed for process control implemented using PLC and intelligent field devices. In the DCS, multipoint and intelligent Network Controlled System (NCS) has replaced the traditional point-to-point architecture which had been used by the industry for decades. The new architecture has smaller volume of wiring and distributed processing resulting into quick and easy maintenance and low cost. This development initiated research in two main directions: (i) Network design and configuration; and communication protocol [7]-[15] (ii) DCS system hardware and software architecture [16]-[19].

The research and development in the area of DCS network design, the configuration and communication protocol is quite active because of the latest developments in the communications field. For example in [7], the authors carried out simulation and experimental evaluation of three control networks commonly used in the industry. The control networks evaluated are: Ethernet, ControlNet, and DeviceNet. Using some case studies, the authors conclude that with shorter or prioritized messages, DeviceNet outperforms the other two networks where as ControlNet is suitable in case of time critical and non-time-critical messages.

In case of Profibus networks, a comprehensive study has been carried out in [8], where authors discuss how Profibus fieldbus networks can be used to support real time communication for industrial systems. The major contribution is that the real time behavior is guaranteed even in the absence of synchronous bandwidth allocation. The author in [9] has performed experimental evaluation of Profibus-FMS. Based on the experimental results, the author suggests an optimal value of target token rotation to reduce the delay of variable messages and an appropriate segment size to reduce delay in domain management service of FMS.

In case of wireless Profibus by integrating wireless and wired devices, a study has been carried out in [10], where the parameters of wireless Profibus protocol standard have been discussed. The author has shown that round robin protocol with three add-ons namely simple data relaying, simple poll relaying, and priority restriction outperforms Profibus protocol under bursty conditions, and if the ring is stable. Normally, this state of keeping logical ring stable in presence of transmission errors is difficult to achieve.

Apart from consideration of network protocols of control networks with associated delay characteristics, a study [11] has been carried out about design consideration for a specific network like DCS. The main evaluation measures for the network quality of service (QoS) are: time delay statistics, network efficiency and utilization, and the number of lost or unsent messages. Based on results, the authors demonstrate that the performance characteristics are useful guidelines for choosing the network and control parameters when designing an NCS.

A different approach to optimize the delay characteristics by use of decreased communications in distributed control systems is suggested and studied in [12]. The authors estimate the outputs at each node of the network and then the expected communication frequency of the system is predicted. The only viability that seems to be of use with this kind of approach is with wireless nodes where transmission errors are relatively high compared to wired nodes, and thus it may be preferable to reduce communications to offset the impact of errors in a situation when relatively higher number of nodes start communicating.

In addition to the well known related works presented above, recently more focus has been reported in literature with respect to distributed intelligence in the network. The automatic reconfiguration resulting in a predictable and stable behavior is another requirement of such systems because if any hardware needs to be added or removed the overall system should not be disturbed and the modification process should remain transparent. This initiated the agent based approach [13]-[14] in the area of DCS. In [13], the authors propose a general approach for dynamic and intelligent reconfiguration of real-time distributed control systems that utilizes the IEC 61499 function block model [16]. The approach takes advantage of distributed artificial intelligence at the planning and control levels. The criterion of handling various delay characteristics that may occur because of environmental or protocol changes in the network have not been addressed in this work, which may in fact simplify the re-configurability of control system itself. Another contribution [14] has been reported, where methodologies and tools for intelligent agents in DCS environment are discussed. This kind of approach helps in building intelligent collaboration of agents, which carry internal and external operational tasks.

The distributed real time control application development for task allocation in heterogeneous systems is also reported in [15]. The approach used in this work uses functional block allocation (FBALL) algorithm [16] that is defined to guarantee real time requirements. This work mainly addresses problems related to real-timeliness in the environment of interoperability. The authors build an application specification that does not involve network considerations or re-configurability of the process itself.

The PLC are the most widely used processing units and fieldbus is most widely used to interconnect the process controllers, sensors and actuators in a DCS system. To realize the plug and play type of operation, the DCS vendors ought to adopt certain standards both for PLC and the communication. The International Electrotechnical Commission (IEC) and PLC manufacturing companies are actively involved in this development and establishment of standards [16]-[19]. These standards define basic concepts for the design of modular, reusable, distributed components for the DCS.

2 Introduction to the proposed approach

The issues addressed in literature cover network considerations, interoperability of devices, customization and reconfiguration of the control system using distributed intelligence, and application specification development. Network speeds available today are in the order of Gbps and with emergence of PROFINET standard, most of the network delay parameters are of less importance. However, the consideration of network parameters in the framework of distributed intelligence makes it a strong candidate for distributed intelligent control system in a situation where network nodes/sensors may be non-stationary, protocol invariant and reconfigurable.

The process under consideration can be visualized by Figure 1. In this Figure, the controller is the main device for which distributed control system is to be investigated. The complete system involves interconnection of control devices, some distributed field and intelligent devices connected to profibus, and few intelligent control devices operating under legacy environment. The control device at lower level of the system may involve standard PLC for motion control of a typical drive, for example. The intelligent devices are normally needed for a specific application, where some of the decisions and local information gathering is done at that level and for onward transmission to the main controller for final network control decision.

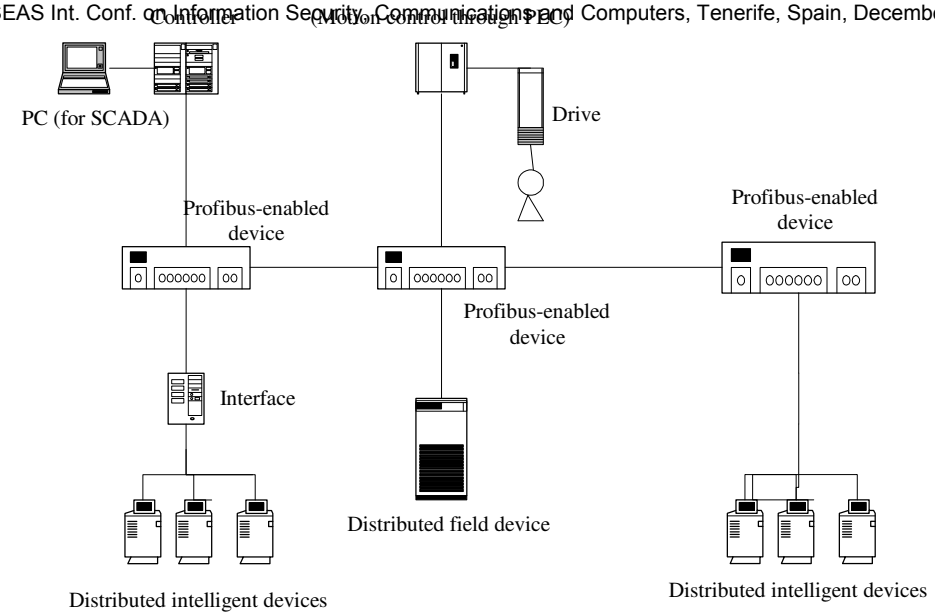

Figure 1: Prceess involving different PLC's and independent devices for distributed network control

As stated earlier, these devices are reprogrammable and reconfigurable to suit changing needs of the process control. The interface shown in the figure is for legacy intelligence devices which are now part of the main distributed control system. The network considered is typical with Profibus compatible switches or network devices with wired or wireless connectivity to a remote device or a group of distributed or intelligent devices. The PC shown is for management purpose, especially for supervisory control and data acquisition (SCADA) system. All the field devices considered are IEC 61804 compliant and PLC are IEC 61131 compliant. The requirements for DCS development for this process control include consideration of reconfigurability and intelligence in the form of framework within a Profibus compatible network.

3 Proposed Approach

Based on discussions in previous section, we discuss our approach for providing a distributive control using an intelligent approach. The idea is to provide an intelligent control using a set of processes. These processes together will help minimize communications between devices and the main controller; and provide a degree of configurability of devices at the same time. The minimization of communications will help thwart any network bottlenecks arising because of interoperability of devices, network protocol change, or simple operational requirements at the device level. The configurability of devices will be provided by collecting operational parameters at the device(s) level, and then estimating of new characteristics of concerned entities at the global level. Using local intelligence collected through interactions and combined with main controller requirements at the global level will help in solving combinatorial complexity present at any time during the operations. Thus, two areas are targeted: social interactions of entities with domain intelligence; and effective decision making set by a main process. The following is the discussion on both processes.

3.1 Local process

Knowledge integration at the global level requires balanced information from local entities. The job of these entities can be done effectively by distributed agents. Agent-based control at the local level can help to handle combinatorial complexity, and provides framework to integrate knowledge [20]-[21]. The agents collaborate socially, learn and adjust their abilities within the constraints of the global process. The functions within this process include: how agents collaborate and clusters are formed to accumulate intelligence and enable decision making; how access to another agent is facilitated; and the procedure for inter-cluster collaboration is established. To make sure that these agents form an intelligent system, the framework has to conform to four requirements [14], that is: the system is decentralized; agents follow centricity; they use common language; and that the system is scalable. In order to highlight distributed agent mechanism, we discuss agent design mechanism and domain ontology in next paragraphs.

Agent design mechanism: There is already more work done on agents alone and details of their definitions, possible roles and life cycle can be found in [22]-[23]. In this work, we do not intend to recreate the work but desire to highlight their framework in the context of customized distributed intelligent control. Agents, as discussed before are active software entities that have social capabilities to interact and can also request for additional capabilities once they discover that the task at hand can not be fulfilled with existing ones. The programming of these agents is done at the global level where a set of heuristics is used for reasoning at the local level, and is stored as a function block diagram (like an internal script). The agents know about their equipment, continuously monitor its state, and thus can decide whether to participate in a mission or not. Thus the design of an agent is not only dependent upon operational parameters of a typical device, but is normally made open ended to add flexibility during operation. Once task is assigned to an agent, it interacts with other agents using a specified mechanism as shown in Figure 2, for example.

The collaborating agents join (on their own will) and thus form a cluster in order to enable a decision making. Within their constraints, they accomplish their task. An agent may collaborate on concurrent clusters, as depicted in Figure 2. These clusters may not exist in any specific location, but are virtual entities of a limited lifecycle performing a common goal. The agents have also limited lifecycle ranging from milliseconds to *n* number of seconds, and can request for an extended life cycle beyond current cluster. In addition to agents, there are other computing units that exist at the local level and which help to form a cluster. These are known as Cluster Directory (CD) and Cluster Facilitator (CF) respectively. The following steps describe the operational scene of agent collaboration in clusters:

- Agent N receives a request from global process for a task planning.
- It checks its internal scripts, if it can participate then it solves the local steps.
- For external steps, it contacts cluster directory to check for other agents if they also have external capability.
- In that case, CD provides contact details.
- Upon receipt of these details, agent N creates CF_N and passes on these details. CF_N understands coordination context.
- CF_N passes the request to specified agents, and thus cluster is formed.

For efficient collaboration, CD must remain updated for recording the information of its members, such as agent name, agent locator, service name, service type, and so on. Upon joining or leaving the cluster, an agent must register or cancel registration respectively through CF. Through a query, an agent can find out other members' services and locators. Through these steps, a trust is developed and thus members hold higher authority than non-members.

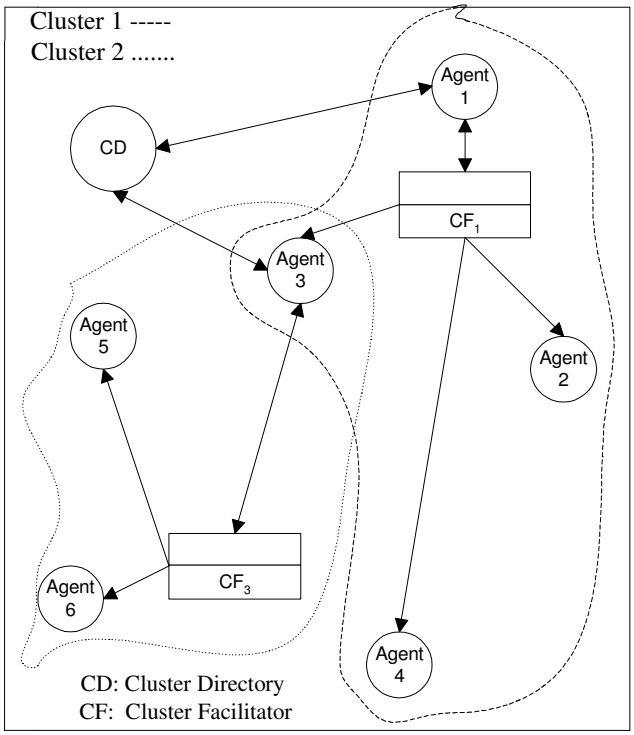

Figure 2: Agent Collaboration

Agent Cluster: Recently developed tools can be used to help design cluster facilitator and domain ontology, using for example DARPA Markup Language (DAML) [24]. The DAML extends XML (Extensible Markup Language) and RDF (Resource Description Framework) to include domain ontology. It provides rich set of constructs to create ontology and to markup information for attaining machine readability and understandability; it has capability of inference so that membership or service of cluster can be precisely defined. In case of XML and RDF, there seems to be no significant literature to manage ontology, and this is why, we have chosen DAML to build cluster ontology. Furthermore, we extend the Foundation for Intelligent Physical Agent (FIPA) agent

management specification [25] to develop the agent role called CF to manage cluster directory and cluster ontology.

Using assistance from DAML-based ontology, the members of the cluster are able to form clusters and communicate with other agents, as shown in Figure 2. The interaction among domain ontology, CD and CF can be best understood using Figure 3. Figure 3 shows how CF gets access to DAML files and facilitates the common goal of the cluster.

There are tools available like Jena semantic web [26] that can be used to handle the cluster director (CD) built using DAML, and to develop a Java class "Directory". From discussions above, the main functions of CD can be summarized, as:

- Add the information of an agent
- Remove the information of an agent
- Get the list of agent names of all members
- Get the information of individual agent by name
- Get ontology used by members in the cluster
- Add external ontology if provided by an agent

Keeping this local process in perspective and using the main functions of CD, the partial directory can be described by Figure 4. The Figure 4 shows information of CF (lines 1-9) and members of cluster (lines 20-22), the cluster directory also records meta-data about cluster such as cluster name (line 12), cluster description (lines 13-15), ontology used in cluster (lines 16-18), and so on.

An Example: Here, we consider a distributed control environment and consider how a distributed field device agent may help to achieve task of a user agent. It should be reminded here that all members' domain knowledge (ontology) may not be the same, and some agents may provide knowledge of the specific field device. In fact, all agents hold basic cluster ontology (the knowledge of the local process) provided by CF, that is, for example, user agent holds basic knowledge of the local process but does not understand the knowledge that a distributed field device holds. Through DAML-based ontology, members can communicate with other to acquire requested service, as shown in Figure 5. It is clear from the Figure 5 that when distributed field device agent joins the cluster, it informs CF about corresponding ontology it provides (Figure 5(a)).

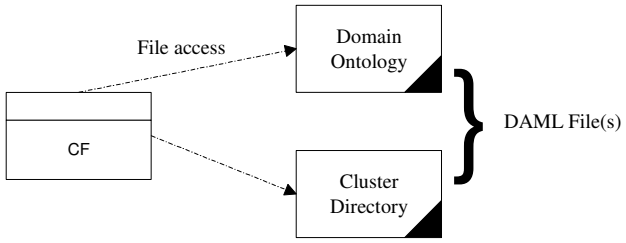

Figure 3. Linking CF with DAML

```
1. <cluster:CF rdf:ID="theCF">
2. <cluster:agentName>"CF"</cluster:agentName>
3. <cluster:agentDescription>
4. "DCS Cluster Facilitator"
5. </cluster:agentDescription>
6. <cluster:locator>
7. "http://dcs.ee.uaeu.ac.ae/DCS/agent/CF"
8. </cluster:locator>
9. </cluster:CF>
10.
11. <cluster:Cluster rdf:ID="DCSCluster">
12.<cluster:clusterName>"DCS"</cluster:clusterName>
13. <cluster:clusterDescription>
14. "Distributed Control System"
15. </cluster:clusterDescription>
16. <cluster ontology>
17. "http://dcs.ee.uaeu.ac.ae/DCS/ontology/dcs.daml"
18. </cluster:ontology>
19.
20. <cluster:hasCF rdf:Resource="#theCF"/>
21. <cluster:consistOf rdf:Resource="#agent1"/>
22. <cluster:consistOf rdf:Resource="#agent2"/>
23. </cluster:Cluster>
```

Figure 4: DCS Cluster Directory

Thus the CF maintains local process ontology plus the distributed field device ontology. This means that ontology can be updated (Figure 5(b)) when ever any agent joins the cluster to perform a common goal. When user agent wants to perform a task, it asks CF about domain ontology and the agents that provide external capability. In response, CF informs the user agent if ontology is to be acquired (Figure 5(c)). Thus, the user agent can communicate with the distributed field device agent (Figure 5(d)).

3.2 Global Process

This process handles core mechanism that glues organization's local process to main global process. This process may handle many functions for example: how library of information collecting agents is to be created; what do these entities do; how do they communicate with each other; how

deadlocks are to be removed; definition of main controller task; estimation of relationships among agents; security of entities at the process level; and above all intelligent decision making after all that information is available. Some of the functions for example library of entities, their job description, definition of controller tasks, and domain ontology can be defined offline before the implementation actually starts. The dynamic components are removing of deadlocks, estimation of characteristics and relationships and intelligent decision making. It seems that all of these dynamic functions together may require high computations, but the advantages gained are many: (i) Reduced communications between main controller and the device(s) to offset any network bottlenecks and provide simplicity to enable better interoperability (ii) Intelligence gathering to build a degree of reconfigurability in case estimated parameters exceed beyond a limit (iii) Reduced human supervision. It can be also argued that complexity of this process is only a technology mismatch, and that if only small scale changes are to be decided at the global level like reconfiguration of device processes, security of agents, then intelligence can be further distributed to the agents at local level.

4 Conclusions

We have presented a customized framework to provide intelligence at device process level using decentralized agents. The agents' communication and cluster behavior is facilitated through cluster facilitator, which updates its domain ontology whenever an agent joins the cluster. The agents are reconfigurable scripts, and their domain parameters are controllable through global process. The idea behind splitting the process into two processes is customization: to maximize decentralization at the local process; to handle reconfigurability at operating system of the main controller. There exists a possibility of introducing intelligence at global level to automate reconfigurability. Currently, we are investigating on these lines, and findings will be presented in future.

References:

[1]. M. G. Ioannides, "Design and Implementation of PLC based Monitoring Control System for Induction Motor", *IEEE Transactions on Energy Conversion*, Vol. 19, No. 3, pp. 469-476, 2004.

[2]. G. Prasad, E. Swidenbank, B. W. Hogg, "A novel performance monitoring strategy for economical thermal power plant operation", *IEEE Transactions on Energy Conversion,* Volume 14, No. 3, pp. 802-809 Sept. 1999.

[3]. R. E. Putman, F.C. Huff, J. K. Pal, "Optimal reactive power control for industrial power networks" *IEEE Transactions on Industry Applications*, Volume 35, No. 3, pp. 506–514, May-June 1999

[4]. M. Buchler, W. Funk, A. Gutierrez, R. F. Harr, P. E. Karchin, P. Liu, S. Nam, J. G. Shiu, S. F. Takach, M. Atiya, D. Padrazo, A. Arefiev, S. Barsuk, F. Khasanov, L. Laptin, V. Tchoudakov, I. Tichimirov, M. Titov, Y. Zaitsev, "Design and operation of front-end electronics for the HERA-B muon detector", *IEEE Transactions on Nuclear Science*, Volume 46, No. 3, Part 1, pp. 126–132. June 1999.

[5]. H. Chang-kun, "A distributed control system of ship diesels", *Proceedings of the IEEE International Conference on Industrial Technology*, Dec. 1996.

[6]. J. Alonso, J. Ribas, J. Coz, A. Calleja, E. Corominas, "Development of a Distributive Control Scheme for Fluorescent Lighting based on LonWorks Technology", *IEEE Transactions on Industrial Electronic*, Vol. 47, No. 6, pp. 1253-1262, 2000.

[7]. F. Lian, J. Moyne, and D. Tilbury, "Performance evaluation of control networks", *IEEE Control Systems Magazine*, pp. 66-83, February 2001.

[8]. E. Tovar, and F. Francisco, "Real-time field bus communications using Profibus Networks", *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 6, pp. 1241-1251, 1999.

[9]. S. Hong, "Experimental Performance Evaluation of Profibus-FMS", *IEEE Robotics and Automation Magazine*, pp. 64-72, December 2000.

[10]. A. Willig, "Polling based MAC protocols for improving real-time performance in a wireless

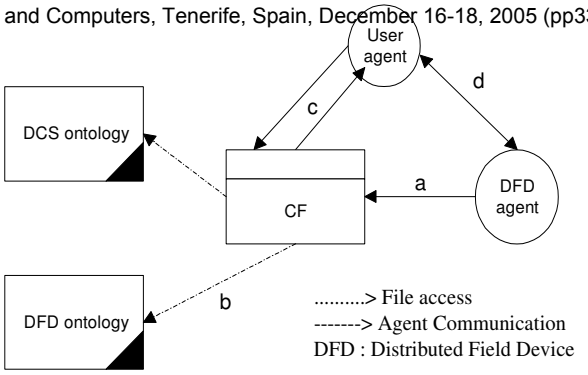

Figure 5: Update in ontology provided by distributed field device agent

Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp33-39)

Prohbas , *IEEE Transactions on Industrial Electronics*, Vol. 50, No. 4, pp. 806-817, 2003.

[11]. F. Lian, J. Moyne, D. Tilbury, "Network Design Consideration for Distributed Control Systems", *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 2, pp. 297-307, March 2002.

[12]. J. Yook, D. Tilbury, and N. Soparkar, "Trading Computation for bandwidth: Reducing Communication in distributed control systems using state estimators", *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 4, pp. 503-518, 2002.

[13]. Bernnan, M. Fletcher, and D. H. Norrie, "An agent-based approach to reconfiguration of the real-time distributed control systems," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 4, August 2002.

[14]. F. Maturana, R. Staron, K. Hall, "Methodologies and Tools for Intelligent Agents in Distributed Control", *IEEE Intelligent Systems*, pp. 42-49, February 2005.

[15]. A. Prayati, C. Koulamas, S. Koubias, and G. Papadopoulos, "A methodology for the development of distributed real-time control applications with focus on task allocation in heterogeneous systems," *IEEE Transactions on Industrial Electronics,* Vol. 51, No. 6, pp. 1194-1206, December 2004.

[16]. Function Blocks (FB) for Process Control, IEC 61804, 2000.

[17]. IEC 61131

[18]. IEC 61131-3

[19]. Function blocks for industrial-process measurement and control systems, IEC 61499, 2001.

[20]. M. Ahmed, "Meeting the Challenges of Complexity", *InTech*, Vol. 51, Oct. 2004, pp.20-23.

[21]. R. Staron, et al, "Use of an Agent Type Library for the Design and Implementation of Highly Flexible Control Systems", *Proceedings of 8th World Multiconference on Systemics, Cybernetics, and Informatics*, 2004, pp. 256-261.

[22]. R. Brennan and D. Norrie, "Agents, holons and function blocks: Distributed intelligent control in manufacturing", *Journal of Applied Systems Studies*, Vol. 2, No. 1, pp. 1-19, 2001.

[23]. M. Wooldridge and N. Jennings, "Intelligent Agents: Theory and Practice", *Knowledge Engineering Review*, Vol. 10, No. 2, pp.115-152, 1995.

[24]. DARPA Agent Markup Language (August, 2000) [Online]. http://www.daml.org/; accessed August 2005.

[25]. Foundation for Intelligent Physical Agents (December 2002), FIPA Agent Management Specification [Online]. http://www.fipa.org/specs/fipa00023/; accessed September 2005.

[26]. HP Labs (January 2003). [Online]. Jena Semantic Web Toolkit Available: http://www.hpl.hp.com/semweb/jena.htm/; accessed September 2005.