An Agent-Oriented Modelling Approach for Agile Manufacturing

Anthony Anosike[1] and David Zhang[1]

[1] School of Engineering and Computer Science,
University of Exeter, Harrison Building, North Park Road,
Exeter EX4 4QF, United Kingdom.
{A.I.Anosike, D.Z.Zhang}@exeter.ac.uk

Abstract. This paper presents a conceptual hierarchically structured multi-agent architecture for manufacturing systems' modelling. The architecture provides a suitable decision-making framework where each agent represents a node in the hierarchic tree of manufacturing systems such as the company as a whole, each plant within the company, each cell or line within the plant, each machine in a cell or line. Each agent has the ability to perceive and evaluate changes that occur in the manufacturing environment, interact with other agents in the system in order to reach an optimal decision, and act based on that decision. In other words, agents respond timely to unexpected changes by continuously co-ordinating their activities, and allocating manufacturing resources dynamically based on actual shop-floor situation. The flexibility of this architecture also lies in its ability to accommodate both homogenous and heterogeneous agents, and its capability for the dynamic addition and removal of agents using a conceptual intelligent communication mechanism similar to the blackboard messaging system.

1 Introduction

The dynamic variations in demand patterns and product mix, driven by unpredictable changes in a global market, are placing manufacturing systems under significant pressure. In order to remain competitive, manufacturing organisations must satisfy demands timely. This implies that companies must increase product varieties, reduce time-to-market, shorten product-life cycles and at the same time maintain good quality and reduce investment costs. Conventional methodologies for planning and control have been found to be inadequate in meeting these challenges. Agile manufacturing provides opportunities for the definition and creation of new concepts and computer-based modelling and control techniques that will assist manufacturers to dynamically and cost-effectively control, configure, adapt, and restructure manufacturing systems to cope with variations in demand patterns and product mix which result from unpredictable market changes. This paper presents a conceptual hierarchically structured multi-agent architecture for manufacturing system's modelling. Multi-Agent Systems are made up of 'intelligent agents'. Unfortunately, there is no universally accepted definition of an agent. This is mainly because of the vast number of research teams in various domains using the same concept to build agents that have different capabilities [1]. As a result, each definition of the term is based on the inherent capabilities of the agent. However, most definitions agree that agents are software units that are designed to accomplish specific tasks on behalf of users and have the following attributes: *Autonomy* (acts without being invoked), *Social Ability* (interacts with other agents by mutual message-based communication), *Reactivity* (agents perceive and respond to changes that occur in their environment)[2].

The proposed multi-agent architecture provides a suitable decision-making framework where each agent represents a node in the hierarchic tree of manufacturing systems such as the company as a whole, each plant within the company, each cell or line within the plant, each machine in a cell or line. Each agent has the ability to perceive and evaluate changes that occur in the manufacturing environment, interact with other agents in the system in order to reach an optimal decision, and act based on that decision. In other words, agents respond timely to unexpected changes by

continuously co-ordinating their activities, and allocating manufacturing resources dynamically based on the actual shop-floor situation. The flexibility of this architecture also lies in its ability to accommodate both homogenous and heterogeneous agents, and its capability of dynamic addition and removal of agents using a conceptual intelligent communication mechanism similar to the blackboard messaging system.

This paper is structured as follows. Section 2 presents the requirements that will be addressed by the multi-agent system. Section 3 provides a background on other emerging distributed manufacturing paradigms such as Holonic Manufacturing Systems (HMS), Bionic Manufacturing Systems (BMS) and Fractal Factory. Section 4 provides an overview of available Multi-Agent System development methodologies. Section 5 provides an overview of some agent architectures. The last section describes the proposed agent architecture that has the capability to fulfil our requirements.

2 Requirements Analysis and Definition

Typically, the life cycles of most systems include requirements analysis, design, implementation and deployment, operation, logistics and maintenance, and decommissioning [3]. Requirement analysis and definition provides a point of focus and ensures that the developed multi-agent system will address stated problems effectively. One of the major problems facing manufacturing enterprises is the inability to respond to changes timely and cost-effectively.

Market changes such as variations in demand patterns, variations in product mix, shorter product life cycles etc. induce further changes to the manufacturing enterprise. Valckenaers et al [4] categorised these manufacturing changes into *Production Change* and *Production Disturbance*. "A Production Change is an alteration to the production condition which is intentionally performed by the plant." This includes the introduction of new products or new product variants, increase (or decrease) of production capacity, introduction of new production technology and changes in the work force. "A Production Disturbance is an unanticipated change to production conditions with negative effect on the process performance." This was further classified into *External and Internal Disturbances.* External disturbances include those caused by customers (such as variations in demand patterns) and those caused by suppliers (such as the ability or inability to deliver raw materials of the right quality and quantity at the right time). Internal Disturbances include equipment failures, quality miss, lack of co-ordination and work force unavailability. In order to respond timely to these changes, manufacturing systems should be *Reactive, Scalable, Flexible, Agile, Self-Motivated, Informative and Self-Organising.*

3 Emergent Distributed Manufacturing Control Paradigms

In recent years, several paradigms have been proposed to enable manufacturing systems adapt flexibly to its ever changing environment. These paradigms include *Holonic Manufacturing Systems (HMS), Bionic Manufacturing Systems (BMS), Fractal Factory and Multi-Agent Systems (MAS)*.

Holons are the basic entities in Holonic Manufacturing Systems. Van Brussel et al [5] defined a holon as "an autonomous and co-operative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects. A holon consists of an information processing part and often a physical processing part. A holon can be part of another holon." The term holon was first coined by Koestler [6] from a Greek word *holos*

(meaning whole) and a suffix *on* (suggesting part as in proton or neutron). While a 'part' implies something fragmentary and incomplete, a 'whole' is considered as something complete. At the same time, *whole* implies that holons are autonomous and *part* implies that holons are part of a community and thus co-operative. A hierarchy of holons could be referred to as a *Holarchy* [6 pp.103]

Bionic Manufacturing models manufacturing systems after the hierarchy of components found in biological systems such as cell, organs, organisms etc. Okino [7,8] proposed a modelon-based approach to model Manufacturing systems. A modelon is composed of *operators and common memory* and a hierarchy of *children modelons.* The operators assist in communication and other interaction between the children modelons. The common memory is the communication medium for the operators and the children modelons. It is a message board similar to the black board system in AI.

Fractal Factory [9] is a concept that originated from the *Theory of Fractal Geometr*y proposed by Benoit B. Mandelbrot [10]. This theory provides tools for analysing and describing geometric objects in multidimensional spaces. The word fractal comes from a Latin word *'fractus*', which means *broken or fragmented*. Each of these fragments or rather, fractals contains the basic characteristics of the whole structure. This idea is believed to provide ways for integrating factory or company operations efficiently [9]. The characteristics of a fractal factory include *self-similarity, self-organisation, self-optimisation, goal-orientation and dynamics*. One of the implications of self-similarity is that each fractal must itself be a (little) 'Fractal Factory'.

Evidently, the above concepts have a lot of overlapping characteristics. This is mainly because they are all striving to achieve the *whole-part* relationships as found in various aspects of nature such as in fractals, living organisms, biological societies. However, the main difference lies in the approach that has been adopted in order to realise the *whole-part* relationships. Agent-oriented technology is an alternative approach for implementing the above concepts. As a distributed computing technique, it is inherently modular and thus can be modelled as fractals, parts, *whole-parts* or *partly-wholes.*

4 Methodologies for the Analysis and Development of Agent-Based Systems

The aim of methodologies is to assist developers in every phase of the life cycle of an agent-based application [11]. These methodologies usually consist of several models, levels or phases. Burmeister [12] describes three models: the *agent model*, the *organisational model* and the *co-operation model*. The agent model defines the agent and its internal structure, the organisational model defines the relationships/responsibilities between agents in their community and the co-operation model describes the interactions amongst these agents.

Kinny et al [13] defines two major levels: the *external* and *internal* viewpoints. The external viewpoint decomposes the system into agents and defines the interactions between them. These are modelled using the agent and interaction models. The internal viewpoint models each BDI (Belief-Desire and Intention) agent class through the belief model, the goal model and the plan model.

Moulin et al [11] defines a methodology that consists of an *analysis* and a *design* phase. The analysis phase consists of *Scenario description, Role functional description, Data & world conceptual modelling* and *System-user interaction modelling*. The Design phase consists of *MAS architecture & scenario description, Object modelling, Agent Modelling, Conversation modelling* and *system design overall validation.*

The methodology proposed by Kendall et al [14] is composed of the following models; *the Function Model, Use Case Model, Dynamic Model* and *the Agent Oriented System* which, in turn, consists of *Agent Identification, Co-ordination Protocols or Scripts, Plan invocation, Beliefs, Sensors, and Effectors.* It is believed that with this methodology, agents can be identified together with their plans, goals, beliefs, sensors and effectors that allow them to deal with objects that exist in the external environment. Collaboration between agents is through scripts and the methodology for the analysis and design of collaboration is based on use cases.

MAS-CommonKADS [15] defines the following models. *Agent Model* describes the main characteristics of the agents, *Task Model* describes the tasks of the agents, *Expertise Model* describes the knowledge needed by the agents to achieve their goals, *Organisational Model* describes both the organisation in which the MAS is to be introduced and the social organisation of the agent society, Co-ordination Model describes the interactions between agents, *Communication Model* describes the human-software agent interactions, and the *Design Model* collects the above models together into a unified framework.

Gaia Methodology for Agent-Oriented Analysis and Design [16] consists of the following models. The *Roles Model* identifies the key roles in the system, the *Interaction Model* describes the links between roles and the associated protocols, the *Agent Model* describes the structure of an agent, the *Services Model* describes the services associated with each agent role and the specification of the main properties of these services and the *Acquaintance Model* describes the communication links between agent types.

5 Agent Architectures

There are three major types of agent architectures: *Deliberative, Reactive* and *Hybrid Architectures* [2]. Deliberative architectures operate in the *sense-plan-act* fashion and are characterised by their goal-oriented behaviour, knowledge representations particularly, knowledge about the world, and the planning process which aims at generating correct and optimal course of action. Non-deliberative architectures operate in a *stimulus-response* fashion. One major characteristic of these architectures is the lack of knowledge representations. In contrast to deliberative agents, which are designed to produce optimal actions, reactive agents produce robust actions [17]. Hybrid agent architectures combine the good qualities of both deliberative and reactive architectures i.e. fast response and the generation of optimal course of action.

6 Proposed Agent Architecture

The agent architecture presented below is believed to provide a mechanism to implement a hierarchic but flexible multi-agent system. This architecture would be developed using the Agent-Oriented development methodology proposed by Burmeister [12] due to its simplicity and ease of implementation. Furthermore, some ideas would be drawn from other sources [8, 18, 19]. The architecture, shown in the figure 1 below, consists of a *control module, common environment and children agents.* The control module consists of the following sub-modules: *inter-agent communication, agent models, knowledge base, and controller*. Inter-agent communication is for communication with other agents in the system and consists of the communication protocols. The Agent Model consists of information about the skills, interests, current status etc. of other agents in the system. The knowledge base contains the *agent's co-operative knowledge* (negotiation, co-ordination and decision methods), *problem-solving knowledge* (procedural knowledge, goals and goal prioritisation methods), and *task knowledge* (primitive and composite tasks and task

decomposition methods). The controller contains the agent's inference, reasoning and control mechanisms, and behaviours. Behaviours represent pre-compiled plans that are triggered upon the occurrence of specific events.

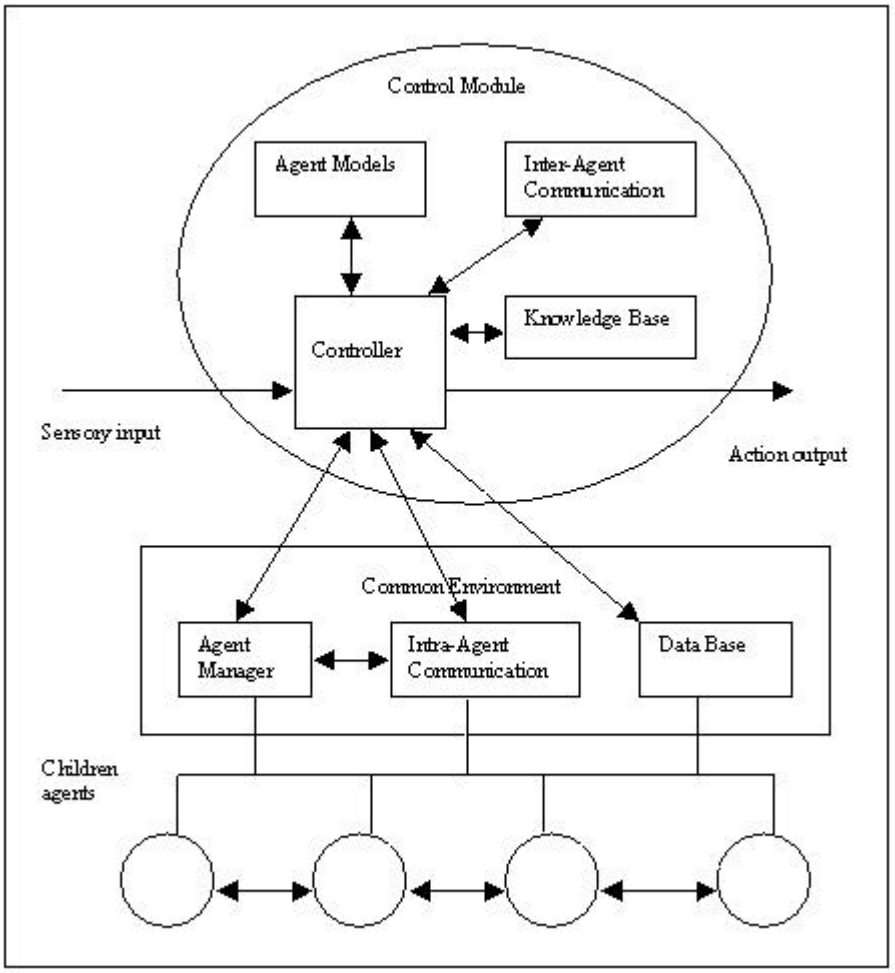


Fig. 1. Agent Architecture

The Common environment consists of the following sub-modules: *database*, *intra-agent communication* and *agent manager*. The database contains detailed information about all the children agents in terms of their capabilities, work-load, current status etc. The intra-agent communication module implements an intelligent communication system similar to message board in Okino [8] and Skarmeas [19], and network facilities in Iglesias et al [18]. This is basically, a blackboard messaging system however, agents do not need to poll the message board. It might be referred to as an intelligent messaging component that informs agents of the changes in the systems such as addition and deletion of agents, service requests etc. However, communication between the children agents is direct between the agents. The intra-agent communication component provides mechanisms for children agents to log in/out of the parent network, register/unregister their capabilities, register/unregister required capabilities. Additionally, it notifies existing agents about new agents that joined the system and also, about agents that were

removed from the system. The manager provides mechanisms for creating, loading, instantiating, stopping and deleting agents and performs theses duties without disrupting other running agents. The agent manager, together with the controller and the intra-agent communication system provide facilities for the dynamic reconfiguration of children agents.

7 Conclusion

Holonic Manufacturing Systems (HMS), Bionic Manufacturing Systems (BMS) and Fractal Factory paradigms have similar characteristics. The most important of these characteristics is '*hierarchical*' organisational structure. From the above discussion, it is evident that hierarchy induces organisational cohesion, which is necessary for achieving global objectives. In view of this, we presented a conceptual multi-agent architecture for the modelling of manufacturing systems. The architecture, which is still under implementation, is hierarchically structured to improve the cohesion among the individual agents.

References

1. Kendall, E.A., Murali-Krishna, P.V., Pathak, C.V., Suresh, C.B.: Patterns of Intelligent and Mobile Agents. Proceedings of the Interantional Conference on Autonomous Agents (1998) 92-99
2. Wooldrigde, M., Jennings, N.R.: Intelligent Agents: Theory and Practice. The Knowledge Engineering Review **10** (2) (1995) 115-152
3. Parunak, V.D., Baker, A.D., Clark, S.J.: The AARIA Agent Architecture: An Example Requirements-Driven Agent-Based System Design. In the Proceedings of the First International Conference on Autonomous Agents (1997)
4. Valckanaers, P.: Analysis and Evaluation of change and disturbances in industrial plants. WP1DisseminationReport http://www.mech.kuleuven.ac.be/pma/project/mascada/welcome.html
5. Van Brussel, H., Wyns, J., Valckanaers, P., Bongaerts, L., Peeters, P.: Reference Architecture for Holonic Manufacturing Systems, PROSA. Computers in industry, **37** (1998) 255-274
6. Koestler, A.: The Ghost in the Machine. London (1967)
7. Okino, N.: Bionic Manufacturing Systems – Modelon-Based Approach. In Proceedings of the CAM-I 18th Annual International Conference, New Orleans, Louisiana, (Computer-Aided Manufacturing – International Inc.) (1989) 485-492
8. Okino, N.: (1992), A Prototyping of Bionic Manufacturing System. In Proceedings of the International Conference on Object-Oriented Manufacturing Systems, Calgary, Alberta, (1992) 297-302
9. Warnecke, H. J.: The Fractal Company. Springer-Verlag, Berlin, Germany (1992)
10. Mandelbrot, B.B.: The Fractal Geometry of Nature. San Francisco. W. H. Freeman (1982)
11. Moulin, B., Brassard, M.: A Scenario-Based Design Method and an Environment for the Development of Multiagent Systems. In Lukose, D., Zhang, C. (eds.): First Australian Workshop on Distributed Artificial Intelligence, Springer-Verlag: Heidelberg, Germany, LNAI Vol.1087 (1996) 216-231.
12. Burmeister, B.: (1996), Models and Methodology for Agent-Oriented Analysis and Design. In Fischer, K. (ed.): Working notes of the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems, DFKI Document D-96-06 (1996)
13. Kinny, D., Georgeff, M., Rao, A.: A Methodology and Modelling Technique for Systems of BDI Agents. In Van Der Velde, W., Perram, J. (eds.): Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World MAAMAW'96, (LNAI Vol. 1038). Springer-Verlag: Heidelberg, Germany(1996)

14. Kendall, E.A., Malkoun, M.T., Jiang, C.: A Methodology for Developing Agent-Based Systems for Enterprise Integration. In Luckose, D., Zhang, C., (eds.): Proceedings of the First Australian Workshop on DAI, Lecture Notes on Artificial Intelligence. Springer-Verlag: Heidelberg, Germany (1996)
15. Iglesias, C.A., Garijo, M., Gonzalez, J.C., Velasco, J.R.: Analysis and Design of Multi-Agent Systems using MAS-CommonKADS. In Intelligent Agents IV (LNAI Volume 1365) Singh, M.P., Rao, A., Wooldridge, M.J., (eds.): Springer-Verlag: Berlin, Germany, (1998) 313--326.
16. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. International Journal of Autonomous Agents and Multi-Agent Systems **3** (3) (2000) 285-312.
17. Ferguson, I.A.: TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents. PhD Thesis, University of Cambridge, UK (1992)
18. Iglesias, C.A.,Gonzalez, J.C., Velasco, J.R.: MIX: A General Purpose Multiagent Architecture. In Wooldridge, M., Muller, J.P., Tambe, M. (eds.): Intelligent Agents II,. Springer-Verlag, LNAI Vol. 1037 (1995) 251—266
19. Skarmeas, N.: Agents as Objects with Knowledge Base State. IC Press (1999)