Multi-Agent Systems vs IEC 61499 for Holonic Resource Control in Reconfigurable Systems

K Kruger, AH Basson*
Dept Mechanical and Mechatronic Eng, Stellenbosch Univ, Stellenbosch 7600, South Africa
* Corresponding author. Tel.: +27 21 808 4250; fax: +27 21 866 155 206. E-mail address: ahb@sun.ac.za.

Abstract
Reconfigurable Manufacturing Systems hold promise to satisfy the requirements of dynamic and competitive modern manufacturing. A holonic control architecture is often used for the control of such reconfigurable systems. In this research, software agents and IEC 61499 function blocks are evaluated as alternative strategies for implementing holonic control for a modular feeder subsystem of an experimental Reconfigurable Assembly System. The strategies are evaluated through four reconfiguration experiments. The evaluation is based on qualitative and quantitative performance measures. The results show that agent-based control is more suitable in this specific case study.

Keywords: reconfigurable; holonic; multi-agent systems; IEC 61499 function blocks

1. Introduction
The modern manufacturing environment is characterized by uncertainty and aggressive global competition, and is subject to change in economical, technological and customer trends [1]. Some of the critical requirements for modern manufacturing systems are [2]: short lead times for the introduction of new products into the system, the ability to produce a larger number of product variants, and the ability to handle fluctuating production volumes and low product prices.

The concept of Reconfigurable Manufacturing Systems (RMSs) is aimed at addressing these needs of the modern manufacturing environment. RMSs have the ability to switch, with minimal delay and effort, between members of a particular family of products by adding or removing (hardware or software) functional elements [3, 4]. The key characteristics of RMSs are [5, 6]: modularity of system components; integratability with other technologies; convertibility to other products; diagnosability of system errors; customizability for specific applications, and scalability of system capacity. RMSs have the ability to reconfigure hardware and control resources to rapidly adjust the production capacity and functionality in response to sudden changes [2, 7].

This paper focuses on the comparison of multi-agent systems (MASs) and IEC 61499 function blocks (FBs) as alternative ways of implementing the controller for a subsystem of an RMS. The next section outlines the relevant controller strategies, followed by a description of the case study used for this research. Some experiments used to compare MASs and FBs, as well as the conclusions, are presented in last sections.

2. Control architectures
System reconfigurability is strongly influenced by the control system's software, hardware and configuration. Traditional manufacturing control systems are typically large, centralized applications [8] which greatly rely on Programmable Logic Controllers (PLCs) [9]. However, traditional control systems do not efficiently satisfy the requirements of modern manufacturing [8] since they require expensive and time-consuming efforts to implement, maintain or reconfigure. "The complexity of the control system grows rapidly with the size of the underlying manufacturing system." [10].

© 2013 The Authors. Published by Elsevier B.V.
Selection and peer-review under responsibility of Professor Pedro Filipe do Carmo Cunha

Keywords: reconfigurable; holonic; multi-agent systems; IEC 61499 function blocks
2.1. Holonic architectures

An alternative to the centralized control architecture for manufacturing is the holonic control approach, in which a holon is an autonomous and cooperative building block for transforming, transporting, storing or validating information [11]. Individual holons have at least two basic parts [12]: a functional component (representing a software entity or a hardware interface represented by a software entity) and a communication and cooperation component (implemented by software).

The implementation of holonic control in manufacturing systems holds several advantages [10]: i) reduced system complexity; ii) reduced software development costs; iii) higher maintainability and modifiability and iv) improved reliability.

Two well known architectures for the implementation of holonic control are PROSA (Product-Resource-Order-Staff Architecture) [13] and ADACOR (ADAptive holonic COntrol aRchitecture) [8]. In ADACOR, the Product holons represent the products available for production, the Task holons represent the production orders and schedules, the Operational holons represent the physical resources and the Supervisor holons are responsible for coordination and optimization.

2.2. Multi-agent systems

Software agents can be used for implementing holonic control architectures [14]. An agent can be defined as a computational system with goals, sensors and effectors, which can autonomously decide which actions to take, in a given situation, to maximize its progress towards its goals [11]. Agents also have the ability to communicate and cooperate with other agents to solve complex problems.

Multi-Agent Systems (MASs) hold several advantages for implementation in RMSs. MASs have high modularity and reconfigurability. The addition or modification of resources can be achieved by simply inserting a new agent into the system or modifying the behavior of an existing agent [11]. Due to their modular and decentralized characteristics, MASs are a way to reduce complexity and increase flexibility in a system [15]. MASs can respond quickly to dynamic changes in the manufacturing environment [16]. Furthermore, agent-based technologies are capable of dealing with autonomy, distribution, scalability and disturbance [2].

There have been several practical implementations of agent-based control for manufacturing control [e.g. 8, 17, 18].

2.3. IEC 61499 function blocks

An alternative control strategy for holonic control implementation is IEC 61499 FBs, which "provides an architectural framework for the design of distributed and embedded control systems" [9] and defines a component-based modeling approach. The goal of IEC 61499 is "to offer an encapsulation concept that allows the efficient combination of legacy representation forms with the new object and component-orientation realities" [4]. An IEC 61499 FB can be understood as an abstraction of a system component, which is implemented and controlled by the FB's software [4]. The event-driven model of FBs adds intelligence and autonomy to the resources of the system, increasing its decision-making ability [19].

A benefit of using the IEC 61499 FBs is that, as a modeling language, it is directly executable and is thus ready for simulation. The simulation model can be seamlessly substituted by the hardware interface to real sensors and actuators. The use of FBs greatly increases the modularity of the system and enables the reusability of software components in the system [9]. FBs also have a robust character which makes it appropriate for implementation in the broader embedded systems domain [4].

The IEC 61499 standard has, however, seen only a few practical implementations, e.g. the automation of a baggage handling system [9], and the first factory installation of an IEC 61499 FB control system [4].

3. Case study

3.1. Physical configuration

The case study used for comparing MAS and FB control implementations was the feeder subsystem of a Reconfigurable Assembly System (RAS) at Stellenbosch University (shown in Fig. 1) [20].

![Fig. 1: Schematic layout of the feeder of the RAS.](image-url)
The feeder is responsible for loading the parts, for a specified product assembly, into fixtures on the pallets that are moved between subsystems by a conveyor. The feeder subsystem comprises reconfigurable singulation units (SUs), machine vision sensors and a pick-and-place robot. The function of a singulation unit is to present one part at a time from a bulk container. The vision system determines whether the part presented by the singulation unit is in a collectable pose, as well as the exact position and orientation of the part. If the part is in a collectable pose, the pick-and-place robot moves the part from the singulation unit to the fixture. The control of the subsystem is entirely PC based.

3.2. Controller architecture

A holonic control architecture based on ADACOR was adopted for the RAS as a whole, in which each physical subsystem (e.g. conveyor, welder, feeder, etc.) is represented by an Operational holon, while the other holons are incorporated into a cell controller.

Since the feeder subsystem comprises separate modules, a holonic control architecture was also implemented for its controller, even though it was itself an operational holon in terms of the RAS. For the feeder controller, the Operational holons included the robot, SUs and cameras. The software parts of the Operational holons were implemented in two levels – Higher Level Control (HLC) and Lower Level Control (LLC). Each module has its own LLC, which provides an interface with the holon’s hardware or hardware-specific controllers. The LLC programs were implemented in C#.

The HLC, which is the focus in this paper, is responsible for decision-making and coordination of the subsystem functions. It also manages communication with both the cell controller and the LLCs, using XML protocols over TCP/IP connections. Both control levels are equipped with XML building and parsing functions. XML protocols were chosen for robustness. The operational instructions to produce a product are stored within a data table and the HLC access this table to accomplish a specified command. The data in the table can either be received from the cell controller or an HMI.

The HLC was implemented both as an MAS and by IEC 61499 FBs. These two implementations are discussed in the following two subsections.

3.3. Multi agent control

The MAS version of the HLC was developed using the JADE (Java Agent Development Environment) and was based on ADACOR, with an agent type associated with each holon type as shown in Fig. 2. The Supervisor agent handles communication with the cell controller and coordinates the feeder’s functions by launching the appropriate Product and Task agents. The Product agent can access all the information required to accomplish the product order from the data table, such as the required task sequence and relevant coordinates. The subsystem hardware actions are then coordinated by the Task agents, through communication with the respective Operational agents. The Operational agents interface with the hardware of the feeder’s subsystem, through the LLCs, and are thus responsible for the execution of hardware actions. This MAS had the following Operational agents: a singulation unit (SU) agent, a Camera agent, a data acquisition device (DAQ) agent and a Robot agent.

During operation, the feeder’s Supervisor agent receives a command from the cell controller when actions are required from the feeder subsystem (e.g. to load the parts of a specific product onto the fixture), and replies with a confirmation message upon completion. When the Supervisor agent receives a command, it launches the appropriate Product agent, which then accesses the relevant information concerning the tasks to be performed - such as the task sequence, and the part and coordinate data.

The Product agent launches the Task agents according to the product information. The Task agents drive the required hardware actions. A Task agent exists for every function inherent in the system, e.g. a specific Task agent is responsible for the loading of one of the required parts onto the fixture.

The Task agents coordinate the Operational agents to perform the desired hardware functions. The Operational agents send the necessary part type and coordinate information to the respective LLC programs. The Operational agents also interact with one another where cooperation is needed to perform a certain hardware function.

As illustrated in Fig. 2, the agents have to communicate to cooperate. This is accomplished using...
the functionality of the JADE agent platform, including the Directory Facilitator and the Contract Net Protocol, through communication based on Agent Communication Language. An ontology was created for intra-agent communication, allowing agents to have a shared understanding of certain concepts inherent in the MAS and specifying which type of manipulation and reasoning can be performed on them [11].

3.4. IEC 61499 function block control

The IEC 61499 FB version of the HLC was implemented using the Java-based FBDK (Function Block Development Kit). The architecture of the FB control was also based on ADACOR, except that the Task holons were not explicitly implemented since their functions are captured in the interconnections in the FB network.

The HLC of the Operational holons of the feeder subsystem are mapped to FB devices. An IEC 61499 device is an abstract model that captures the information-processing properties of control devices. These devices are hosts to resources, which in turn contain the FB networks [4]. The FB networks are where the control algorithms are implemented.

The feeder subsystem devices are (Fig. 3): FB_SUPERVISOR, COMMAND_EXECUTE, SU, DAQ, CAMERA and ROBOT.

![Fig. 3: IEC 61499 device network.](image)

FB_SUPERVISOR contains a FB network which handles communication with the cell controller. It contains an XML_builder and an XML_parser FB, which use the standard Java functions for building and parsing XML strings.

The COMMAND_EXECUTE device receives the data from the FB_SUPERVISOR device when the latter’s appropriate output event is triggered. The COMMAND_EXECUTE device contains an FB network for selecting a product and an FB network for each product. The product FB networks are responsible for the coordination of the tasks to perform the received command. This coordination is achieved through the triggering of specific output events. The task sequence, as well as associated part and coordinate information, are retrieved from the data table.

The SINGULATION_UNIT, DAQ, CAMERA and ROBOT devices represent the respective HLCs of the Operational holons. These networks handle coordinates, part types and the functions which are related to specific input events. The networks use XML manipulating FBs to communicate with the respective LLCs.

4. Reconfiguration experiments

RMSs are aimed at situations where changes to the production requirements are to be expected. MAS and FB control strategies were therefore compared by means of four reconfiguration experiments. Consider the key characteristics of RMSs [21] — convertibility and customizability were assessed in the first three experiments, through changes to a product’s processes and the introduction of a new product. In the last experiment, modularity, integratability and scalability are assessed by adding new hardware to the feeder.

The reconfiguration experiments used quantitative and qualitative measurements. The quantitative measurement is implementation time, which comprises development time (time spent on software development before introduction to the system) and reconfiguration time (time required to implement the changes with the system offline). The experiments were performed by the original software developer, with the time measured for the reconfiguration and development periods.

The qualitative measurements correspond to four of the characteristics of RMSs, i.e. convertibility, integratability, scalability and customizability. The explicit measurement of modularity and diagnosibility is omitted, as modularity is already inherent in the design of the control strategies and diagnosibility is highly platform-specific.

4.1. Experiment 1: Change in task sequence

This experiment involved the changing of the sequence in which tasks are performed to load a product sub-assembly into a fixture. The ease by which such a reconfiguration is achieved points towards the customizability of the HLC.

The sequence of tasks was changed in the data table used by the HLC. For both the control strategies, the reconfiguration can be done without the change of any HLC code. For the MAS, the sequence of the tasks involved in a product is controlled by Product agents, which directly implement the sequence as obtained from the data table. Similarly, in the FB HLC, the FB networks of the PRODUCT resources access the data table for the required sequence information.
Since both control strategies are Java-based, data can be accessed easily from a data table which allows both strategies to be equally customizable, since the product data was not built into the software. The reconfiguration requires no implementation time, since no changes had to be made to HLC code (Fig. 4). The result would be similar if the experiment entailed a change to the coordinates involved with an existing task.

4.2. Addition of product tasks

A new task was added to an existing product in this experiment, i.e. the placement of an additional part into the product fixture.

This reconfiguration entails the addition of a new Task agent to the MAS. The Task agent must contain the information regarding the necessary actions to perform the task, such as communicating with the Operational agents and handling the relevant coordinate and part information. The IEC 61499 FB control strategy requires the alteration of the COMMAND_EXECUTE device. The FB network of the PRODUCT resource must be adjusted to incorporate the execution of the newly added task. The new task is entered into the data table, along with the relevant part and coordinate information. This reconfiguration requires a change to the HLC in both control strategies. The level of complexity of these changes reflects the convertibility of the strategies. The implementation has shown that the FB control system requires less development time than the MAS. This is due to the increased complexity of the Java agent code of MASs. The MAS requires no reconfiguration time (Fig. 4) as the changes can be made during subsystem operation. The FB HLC appears as slightly more convertible in this case, but lacks the ability to make online changes like the MAS.

4.3. Addition of new product

This experiment involved the addition of a new product to be handled by the feeder subsystem. A new product entails a new combination of product tasks, with different parts and coordinates. The HLC’s convertibility and customizability is reflected by this experiment.

For the MAS, a new Product agent must be developed. This agent must retrieve the relevant information from the data table and initiate the appropriate Task agents to accomplish the loading of the new product. In the FB implementation, a new product resource must be added to the COMMAND_EXECUTE device. The FB network of this resource must also retrieve the relevant part and coordinate information from the data table and incorporate all the necessary communication channels to accomplish the loading of the product. The construction of both the new Product agent and PRODUCT resource can be done offline.

The experiment shows that the intra-FB communication limits the convertibility and customizability of the HLC. The lack of flexibility again requires the alteration of connections between FBs with the addition of a new PRODUCT device - this difficulty propagates through the coordination and execution of the task sequence. The distribution of information and decision-making ability, along with the communication interfaces, allow for the easy introduction of a new Product agent to the MAS. The changes to the MAS can again be implemented online, as is shown in Fig. 4.

Fig. 4: Summary of experimental results.

4.4. Addition of new hardware

In this experiment a new singulation unit was added to the feeder subsystem. This singulation unit, which is already accompanied by an LLC program, must be controlled and utilized by the HLC. This experiment evaluated the scalability and integratability of the HLC.

The introduction of the new hardware means that a new agent must be added to the MAS and a new device to the FB HLC. In the MAS a new SU agent is added. The agent was programmed to utilize the existing ontology for intra-agent communication and the XML functionality to communicate with the LLC. For the FB control system a new SU device was added, which contains a FB network with the appropriate communication and decision-making functionality. The new HLC programs can again be constructed offline and introduced once ready.

The scalability of the system is reflected in the capacity increase with a hardware addition. This is more easily achieved with MAS than FB control. This is due to the functionality of the Directory Facilitator, which allows for the seamless introduction of agents to the system. The new agent can be utilized without any additional programming. The corresponding introduction to the FB HLC required some alteration to the FB networks. The integration of the existing system with the
new hardware was done with similar ease for both strategies due to their inherent modularity. The implementation time measured for this experiment is also shown in Fig. 4.

5. Conclusions

This paper presents research into multi-agent systems (MASs) and IEC 61499 function blocks (FBs) as possible control strategies for implementation in reconfigurable systems. The control strategies were implemented in the feeder subsystem of an experimental reconfigurable assembly system. The strategies were then evaluated and compared through reconfiguration experiments.

The experimental results show that MASs required less effort and time to implement reconfiguration changes than the FBs in three of the four experiments. This is predominantly due to the flexible communication abilities of the MAS. The FBs have inherent simplicity - because of the visual distribution and connection of information and decision-making logic - but is limited by the lack of intra-FB communication flexibility. The result is that, even though the changes to be made are not very complex, they are time-consuming. The fact that changes can be made to the MAS with the system being online has great advantages regarding reconfigurability.

The two control strategies are both suitable for implementation in reconfigurable systems. Even though MASs appear more suitable in this study, both strategies still require development. The issue of diagnostics requires more research, but the authors expect an MAS to be more challenging to diagnose due to the non-deterministic nature of its operation. However, the currently available FB development environments are far less mature than the MAS platforms, which also greatly increase diagnosing efforts.

Acknowledgements

The funding support of the Technology Innovation Agency (grant AMTS-07-10-P) and the technical support of CBI Electric: Low Voltage are gratefully acknowledged.

References

[1] Leitao, P. and Restivo, F.J., 2006. ADACOR: A Holonic Architecture for Agile and Adaptive Manufacturing Control. Computers for Industry, Vol. 57, No. 2: 121-130
[2] Bi, Z.M., Lang, S.Y.T., Shen, W. and Wang, L., 2008. Reconfigurable Manufacturing Systems: The State of the Art. International Journal of Production Research. Vol. 46, No. 4: 967 - 992
[3] Martinsen, K., Haga, E., Dransfeld, S. and Watterwald, L.E., 2007. Robust, Flexible and Fast Reconfigurable Assembly System for Automotive Air-brake Couplings. Intelligent Computation in Manufacturing Engineering. Vol. 6
[4] Vyatkin, V., 2007. IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design. North Carolina: Instrumentation, Systems and Automation Society, ISA
[5] Mehrabi, M.G., Ulsoy, A.G., Koren, Y., 2000. Reconfigurable Manufacturing Systems: Key to Future Manufacturing. Journal of Intelligent Manufacturing. Vol. 13: 135 - 146
[6] EIMaraghy, H., 2006. Flexible and Reconfigurable Manufacturing System Paradigms. International Journal of Flexible Manufacturing System. Vol. 17:61-276
[7] Bi, Z.M., Wang, L. and Lang, S.Y.T., 2007. Current Status of Reconfigurable Assembly Systems. International Journal of Manufacturing Research, InderScience. Vol. 2, No. 3: 303 - 328
[8] Leitao P. and Restivo F.J., 2008. Implementation of a Holonic Control System in a Flexible Manufacturing System. IEEE Transactions on Systems, Man, and Cybernetics. Vol. 38, No. 5: 699 - 709
[9] Black, G. and Vyatkin, V., 2009. Intelligent Component-Based Automation of Baggage Handling Systems with IEC 61499. IEEE Transactions on Automation Science and Engineering. Vol. 7, No. 2: 337 - 351
[10] Schönz-Reiter, B. and Freitag, M., 2007. Autonomous Processes in Assembly Systems. Annals of the CIRP. Vol. 56: 712 - 730
[11] Paolucci, M. and Sacile, R., 2005. Agent-Based Manufacturing and Control Systems. London: CRC Press
[12] Kotak, D., Wu, S., Fleetwood, M. and Tamoto, H., 2003. Agent-Based Holonic Design and Operations Environment for Distributed Manufacturing. Computers in Industry. Vol. 52: 95 - 108
[13] Van Brussel, H., Wyna, J., Valkenaers, P., Bongaerts, L. and Peeters, P., 1998. Reference Architecture For Holonic Manufacturing Systems: PROSA. Computers in Industry. Vol. 37: 255 – 274
[14] Monostori, L., Vaneza, J. and Kumara, S.R.T., 2006. Agent-Based Systems for Manufacturing. Annals of CIRP. Vol. 55:697-720
[15] Vrbô, P., Lepschütz, W., Valicke, M., Merdan, M., Resch, J., 2009. Integration of a Heterogeneous Low Level Control in a Multi-agent System for the Manufacturing Domain. 2009 IEEE International Conference on Systems, Man and Cybernetics. pp. 7 - 14
[16] Xie, H., Shen, W., Neelamkavil, J., Hao, Q., 2007. Simulation and Optimization of Mixed-model Assembly Lines using Software Agents. Proceedings of the 2nd international conference on Changeable, Agile, Reconfigurable and Virtual production. pp. 340 - 347
[17] Candido, G. and Barata, J., 2007. A Multiagent Control System for Shop Floor Assembly. Proceeding of the 3rd International Conference on Industrial Applications of Holonic and Multi-agent Systems. pp. 293 - 302
[18] Marik, V., Vrbô, P., Tichý, P., Hall, K.H., Staron, R.J., Maturana, F.P., Kadera, P., 2010. Rockwell Automation's Holonic and Multiagent Control Systems Compendium. 2010 IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and reviews. Vol. 41, No. 1: 14 - 30
[19] Wang, L., Cai, N., Feng, H.Y., 2007. Dynamic Setup Dispatching and Execution Monitoring using Function Blocks. Proceedings of the 2nd international conference on Changeable, Agile, Reconfigurable and Virtual production. pp. 699 - 708
[20] Kruger, K., 2013. Control of the feeder for a reconfigurable assembly system. MSc Thesis. Department of Mechanical and Mechatronic Engineering, Stellenbosch University
[21] Wiedahl, H.P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiedahl, H.H., Duffie, N. and Brieke, M., 2006. Changeable Manufacturing - Classification, Design and Operation. Annals of CIRP. Vol. 55:783-809