Multi-Agent Architecture for Self-Configuring Modular Assembly Systems

Pedro Ferreira*, Niels Lohse*
Svetan Ratchev*

*The University of Nottingham, Precision Manufacturing Centre, Nottingham, NG7 2RD
United Kingdom (Tel: +44 115 951 3875; e-mail: epxf@nottingham.ac.uk).

Abstract: Assembly systems today require an increasingly higher level of physical and logical adaptability. Methods and tools are needed to adapt assembly systems to market changes in a timely and cost effective manner. The requirement for increased responsiveness has led to the development of new modular concepts which provide the bases for achieving higher system adaptability through increased component/module interchangeability and reusability. The modularization of physical and control infrastructure does, however, only address one aspect of the issue and there is still a lack of appropriate tools and methods to support the rapid configuration and re-configuration of such systems for changing sets of requirements. This paper proposes an agent architecture for the self-configuration and re-configuration of equipment modules into systems based on a given set of requirements. A new agent model has been developed which addresses the specific needs of modular assembly systems catering both for physical and logical constraints of the modules as well as their joint emergent behaviour.

Keywords: Agent Architecture, Assembly System Design, Requirements Driven Configuration

1. INTRODUCTION

The question of component reusability, rapid configuration and reconfiguration to enable “Configure to order” of assembly systems has become increasingly more important due to ever decreasing product life-cycles and rising process complexity. General purpose assembly machines, equivalent to CNC machine tools, are only available in specialist domains such as printed circuit board assembly where the components are highly standardised. The assembly of most other products demand custom made systems which address the specific requirements for these products. Today, these are mostly “Engineered to Order” making them cost and time intensive to design or reengineer. Increased modularisation of assembly equipment, rapid integration and design tools are considered fundamental for the move towards cost and time effective configuration and re-configuration of complex assembly systems (Koren et al., 1999, Kratochvil 2005, and Onori at al., 2002).

Significant effort has been directed towards creating modular assembly system (MAS) architectures for physical equipment and control interchange ability. The EU project EUPASS has created a framework for rapid integration for ultra-precision assembly modules defining hardware interfaces, control interfaces, and module description formats (EUPASS). Several other modular assembly system platforms have been proposed (Alsterman and Onori, 2001, Gaugel et al., 2004, Hollis and Quaid, 1995). While the number and completeness of underlying industrial applicable standards is still limited, there is a clear drive to overcome this barrier.

Standardisation of hardware and software interfaces is, however, only one aspect of rapid assembly system configuration. Effective tools and method for the requirements driven selection, integration and validation of complex assembly system solutions are also needed to drastically reduce the time and effort required for the development of highly dedicated assembly systems. Most configuration methods reported today adopt a top down approach providing different either methods for stepwise decomposition of the given set of requirements and subsequent solution synthesis or methods for the adaptation of similar system solutions. These approaches are often limited in their scalability and extendibility making them inappropriate and too complex for most MAS configuration problems.

This paper proposes a multi-agent architecture for the self-configuration and re-configuration of MAS based on product and process requirements. The proposed architecture is directed towards utilising latest agent-based negotiation protocols to enable the scalable and extendable requirements driven configuration of complex MAS from a library of existing modular building blocks. In the following section the relevant literature has been reviewed clearly identifying the expected advantages of the proposed approach and existing knowledge gaps which need to be addressed. This paper builds on the concepts devolved in EU project EUPASS and targets de enhancement of the Evolvable Assembly Systems (EAS) concept. The specific characteristics of the MAS configuration problem are defined in section 3. Section 4 gives a detailed definition of the proposed multi-agent architecture. The underlying approach is illustrated with an example in section 5 and finally first conclusions and future work are outlined in section 6.
The suitability of different methodologies for configuration design depend on the complexity of the system (Bi et al., 2008). Systems can be classified as an uncoupled system, loosely-coupled system, or strongly-coupled system. In uncoupled or loosely-coupled systems the components can be determined individually based on their corresponding requirements. This might require some adjustments to individual components. MAS are normally comprised of strongly-coupled and loosely-coupled systems. Configuration methodologies for MAS consequently need to be able to provide solutions for both.

Configuration design of MAS is comparable to the design of modular products (Bi et al., 2008). Therefore there are many methods that can be applied such as feature-based methods (Perreman, 1996), hierarchical decomposition methods (Tsai and Wang, 1999), combinatorial synthesis method (Levin, 2002), entity-based methods (Hong and Hong, 1998), and case-based methods (Watson, 1999).

In strongly-coupled systems the all design variables have to be considered together to validating if the configuration fulfills its requirements (Bi et al., 2008). Early works have focused on sequential design procedure and most of them have considered only one aspect of the system behaviours (Chen and Burdick, 1995, Paredis and Khosla, 1993). However, the coupling of design variables suggests that a concurrent design approach would be better suited to finding global optimal solutions (Bi and Zhang, 2000). Concurrent design, however, can increase the problem dimension which increases the computational effort.

The configuration design at system level is usually achieve through a propose-and-test approach in conjunction with system simulation software for testing. An approximate solution is found in a time-consuming iterative process. Mathematical formulation for the system level would be too complex and it is usually only used for specific sub-problems (Bi et al., 2008). Deterministic models where the system variables are constant have been used in configuration design (Tang L., 2004) these does, however, limit the flexibility of the design system. Stochastic models arise as a solution to this problem since they provide at least one uncertain variable. Some configuration design methodologies have used stochastic models in order to deal with the configuration problems (Ohiro et al., 2003, Zhao et al., 2000).

Despite the significant work in the area of configuration methods, there is still a lack a clear configuration design methodology which is applicable beyond the boundaries and assumptions of specific problem domains. Furthermore, most of the proposed methods in this field of manufacturing system configuration has been focused on the machine level, while the systems have been designed largely intuitively (Bi et al., 2008).

Agent technology is seen as the natural way to address the problems of scalability and flexibility manufacturing systems. As a result agent technology has been widely applied to provide solutions in the manufacturing domain (Shen et al., 2006). This provides extensive literature in agent models, negotiation models, agent environments, etc, however these models are mostly application specific.

One of the key advantages of agent technology is its adaptability, which enables it to be applied to different levels guaranteeing an overall integration. Agents have been used to represent physical manufacturing resources (e.g., machines, robots, tools, fixtures, etc.), aggregations of such resources and even operations of such resources (Shen et al., 2006). This suggests that agent-based architectures are well suited to support manufacturing systems design. The CoBASA targets fast integration and shop floor adaption of assembly systems particularly focusing on the control of such system (Oliveira, 2005). The MetaMorph I and II for instance is an agent-based architecture for distributed intelligent design and manufacturing with the objective of integrating the manufacturing activities (e.g., design, planning, scheduling, simulation, execution, etc) with the activities of the suppliers, customers and partners within a distributed system (Shen et al., 2000). The AARIA agent architecture is another example which also provided an agent-based system design presenting another agent organizational and collaboration model which was more requirements driven (Parunak et al., 2001). Although this project describes a requirements driven approach, it is very case specific which significantly restricts its applicability to other problems domains.

The exiting agent-based approaches are mainly focussed on providing agility and reconfigurability. For configuration design, they also need to have the tendency to provide close to optimal solutions. The optimization in distributed systems is quite different from other approaches which target global optimization through mathematical formulation of the whole problems. The mathematical formulation of complex system is quite difficult to develop and maintain, thus are mainly successful for simpler systems. Agent approaches on the other hand attempt to achieve optimization through efficient coordination mechanisms and thus require significantly simpler models (Shen et al., 2006). Despite all the work done in this field there is still no agent architecture which focuses on the specific issues of modular assembly systems identified by Lohse 2006, namely providing clear formalisms for equipment capability and interconnection constraint representation as well as methods and protocols for system formation from these modules based on a given set of requirements.

3. PROBLEM DEFINITION

Currently the design of assembly systems is a human driven approach based on the expertise of system integrators. Although this process provides valid system configurations, it can be quite time consuming, often considers only a fraction of the possible solution space, and does seldom provide repeatable and transparent solutions. The MAS paradigm with its focus on clear functional decoupling of equipment module functionalities and standardised interfaces for interchange ability has opened the scope for automatic configuration methods. It becomes possible to clearly formalise the functional capabilities and connectivity
constraints of the available modules hence allowing the mapping of required against available capabilities. The design of MAS is therefore essentially a conjoint equipment and process configuration problem at several levels of granularity with equipment modules and their functional capabilities (skills) as the elementary building blocks.

The MAS configuration problem can be defined as illustrated in Figure 1. A set of product, process and business requirements needs to be translated into possible assembly system solutions using a given set of equipment modules. A set of methods and tools will be required to determine both the technical and logical completeness of different configurations and establish their respective performance characteristics. One of the key challenges is the concurrent solution configuration for both the process logic based on the available skills of equipment modules and the physical hardware required to execute the process logic. Inconsistencies in the process configuration can lead to equipment changes and vice versa.

![Fig. 1. MAS Configuration Problem Overview](image)

The number of possible combinations of modules required for an assembly system solution depends on the number of available modules, their connection constraints and the complexity of the given assembly process requirements. This becomes very large even for relatively small problems making configuration based on exhaustive enumeration practically infeasible. An appropriate MAS configuration methodology needs to more goal oriented. Furthermore, any method should be able to exploit the specificities of the MAS configuration problem to reduce the search space. Most MAS solutions above a very low level of complexity can be split into a number of loosely coupled sub-problems with corresponding solutions making them hierarchical in nature. Additionally, elementary equipment modules often have specific predefined roles within a solution and can be classified accordingly reducing the possible number of candidates for specific aspect of a configuration.

While heuristic search and linear programming methods are able to solve these kinds of configuration problems, they require quite complex models and are difficult to maintain. Also they apply a top down approach which only takes limited to no advantage of the hierarchical nature of the problem. Consequently we propose an agent architecture for the bottom up solving of this configuration problem maximising the parallel computation and taken advantage of latest negotiation protocols to achieve a goal oriented behaviour of the overall configuration environment.

4. AGENT ARCHITECTURE

This section gives a detailed overview of the proposed agent-architecture for MAS configuration and explains the rational behind it. The GAIA methodology developed by (Zambonelli et al., 2005) has been applied to translate the requirements identified for MAS configuration problems above into an appropriate agent architecture. A schematic overview of the resulting agent architecture is given in Figure 2.

The clear common denominator of all configuration design methodologies including for MAS is the need to elicit and maintain the system requirements independent of the proposed solution alternatives. Consequently there is a need for Requirements Agents which are able to provide clear objectives to those agents involved in the configuration process. Furthermore, they need to be able to represent the interests of the customer/system user to validate possible system configurations against the original requirements.

Another important set of actors within this problem domain are the equipment modules. It is proposed that each equipment module should be represented by an Equipment Module Agents that have a detailed understanding of the module’s capabilities and behaviour. The Equipment Module Agents have to play a key role in the bottom up approach to the MAS configuration process which will lead to equipment modules begin able to self-configure themselves. Each Equipment Module Agent should only aware of its own capabilities and should only a very limited understanding of the surrounding world to maximise the adaptability and scalability of the framework. Consequently there needs to be a mechanism which validates the logical consistency of the agreed interactions between collaborating Equipment Module Agents. In case of MAS configuration the interactions between modules are two fold: physical connection of the equipment and logical interrelations of their process capabilities. The role of mediator has been assigned to two types of agents as the configuration of MAS always needs to address both the equipment and the process: the Physical Broker Agent and the Assembly Process Broker agents. Both are specialised to validate the logical constraints of physical and assembly process configurations respectively. They also need to be able to negotiation the constraints which exist between equipment configuration and process configuration. Furthermore, they need to be able to facilitate the fine-tuning the configuration based on cost, time, accuracy, and flexibility and learn from past experience to improve future configurations.

This architecture will potentially provide a very large number of possible solutions. Some method for early evaluation of the likely success a consortium needs to be available to reduce the computation effort required. Ideally this evaluation should be synthesised from the actual performance
characteristics of the modules. To provide some bases for early comparison, it is proposed that the Equipment Module Agents deploys Physical and Assembly Process Agents into a simulation environment. They represent the physical and process capabilities of the modules and dynamically interact with each other to determine the emergent behaviour and performance characteristics of a consortium.

The organizational structure for these agents is based on the agent-roles discussed above. The Requirements Agent is hierarchically above the Equipment Module Agent, since it triggers the beginning of the collaboration process and it terminates it by making a selection. The Assembly Process Agent and Physical Agent are children of the Equipment Module Agent, since this agent is the only one with the information required to deploy them. The Physical Broker Agent and The Assembly Process Broker Agent are on higher level from the Equipment Module Agent since they can provide a global view of the configurations. All agents have to be able to speak with agents of the same type plus with agents for which interactions were described.

4.1 Requirements Agent

Requirements Agents are responsible for eliciting the assembly system requirements from a system integrator and advertising them in an understandable format to the other Equipment Module Agents. This agent will provide the assembly tasks description to the interested agents, which entail the basic requirements for the system. The Requirements Agent has also the capability to access the established collaborations to select the most suitable assembly system. This selection is based on the relevant assembly system aspects, namely cost, time, quality/precision and flexibility. Finally, this agent can also negotiate with the system integrator some trade-offs between the established systems constraints and requirement.

4.2 Equipment Module Agent

Equipment Module Agents represent the equipment modules containing detailed models of their connection constraints in terms of available interfaces, capabilities in terms of skills, and behaviour information terms of geometric, kinematic and dynamic models. The Equipment Module Agent’s main objective is to participate in as many successful configurations as possible. It constantly monitors the adverts for new system requirements to identify opportunities for its own set of capabilities. Once the agent identifies an opportunity to participate in a system configuration, it proactively engages in negotiation with other Equipment Module Agents to establish a collaboration which will fulfill the given set of system requirements. The basis for negotiation is the individual capabilities of the Equipment Module Agents and their expected contribution to the success of the consortium. The Equipment Module Agent needs to find other equipment module agents that are willing to collaborate to fulfill the set of requirements. Once a potential collaboration is identified these agents will find a Physical Broker Agent and an Assembly Process Broker agent to mediate the agreement between the members of the consortia, identify conflicts and missing requirements. Once collaboration is finalised this agent will deploy the Physical and Assembly Process Agents into the virtual simulation and validation environment. This agent will also be able to interact with their counterparts and analyse the technical validity of a given configuration and its expected performance characteristics.

Fig. 2. Agent-Architecture Overview
4.3 Assembly Process Broker Agent

Assembly Process Broker Agents act as assembly process configuration experts. It has a global understanding of the assembly process integration and its constraints. These agents are able to determine the logical completeness and correctness of assembly process configurations agreed by an associated consortium of Equipment Module Agents. Its role is to inform the consortia of any missing assembly processes steps. An Assembly Process Broker Agent will advise the consortium of the requirements to fulfilling the identified gaps in the configuration while fine tuning it. The agent will determine these missing elements based on his knowledge of the assembly processes and also through the collaboration with the Physical Broker Agent who plays a similar role in terms of physical characteristics.

4.4 Physical Broker Agent

Physical Broker Agents act as physical configuration experts. This is achieved through a global understanding of physical system configurations and their constraints. The Physical Broker Agents go beyond a mere plugability analysis, looking at the configuration as a whole and determining missing physical connections which would prevent the modules to work together while fine-tuning the configuration. To enable such capabilities, this agent will possess a geometric understanding of the equipment modules and their motion capabilities. All of this is done in close collaboration with the Assembly Process Broker Agent since changes in either hardware or process configuration has an impact on the other.

4.5 Physical Agent

Physical Agents represent the physical behaviour of the module and can in collaboration with other Physical Agents to simulate the joint physical behaviour of a configuration. These agents will collaborate with the Assembly Process Agents for a more realistic simulation.

4.6 Assembly Process Agent

Assembly Process Agents represents the process capability of an equipment module and contains a description of its behaviour. This enables these agents to participate in the simulation of a proposed assembly process configuration in close collaboration with the Physical Agents to establish the overall joint behaviour and performance of the process.

5. ILLUSTRATIVE EXAMPLE

This section shows the application of the proposed agent architecture with the help of an illustrative example which is both simple enough to follow and complex enough to show the potential of the architecture. Let us assume that an automatic workstation for the gluing of two components is required. The user or system integrator would start a new Requirements Agent and specify the desired characteristics for the joining of the two components. At the centre of this would be the process requirements which in this case would consist of a minimum of three operations: apply glue, place one component on the other, and cure the glue. Each would have a set of required process characteristics such as type of glue, amount of glue, positioning accuracy, and curing time. They would also have precedence constraints between each other specifying the order in which they need to be carried out.

Let us further assume that there are a number of Equipment Module Agents available which could deliver some parts of the required process. Take for instance Manipulator B. This module is able to place the two components onto each other but does not have the capability to apply glue. The module has to look for a collaborator who could help deliver the whole process, say the Glue Dispenser. If they decide to collaborate they would contact an Assembly Process Broker Agent and a Physical Broker Agent to facilitate their joint configuration of the assembly process and hardware. The Assembly Process Broker Agent would immediately identify that some essential operations for an automatic process are missing including the supply of the incoming components and the transport of the finished assembly. This will lead to for instance an additional Conveyor module to be included in the consortium. Simultaneously the Physical Broker Agent would identify that both or indeed all three modules could not possibly be connect together without a Base Table to connect them. After a few iterations a logical and physical valid configuration would be established or the consortium would

![Fig. 3. Example of Alternative MAS Configurations](image-url)
be dissolved. In the former case, the joint performance of the configuration would be tested by each Equipment Module Agent deploying a representation of itself (Physical Agent) and its required capabilities (Assembly Process Agents) into a simulation environment where they communicate with each other to establish the overall behaviour and performance of the system. The performance of the collaboration is continuously measured against the requirements set by the Requirements Agent. This again can result in either successful completion of the configuration process or required changes in the consortium. Other collaboration could be formed while this collaboration was put together resulting in competing consortia as only one system is needed (see figure 3).

6. CONCLUSIONS

The presented agent architecture shows clear potential to enable the self-configuration of MAS. This is expected to provide a better support for MAS and faster configurations that provide better results. This architecture also supports the scalability of MAS providing the necessary methods to introduce new equipment modules as well as enhance the overall knowledge of this agent environment. Furthermore, the proposed architecture is also well suited to realise self-planning and adaptation with distributed assembly system control environments which significantly expends the scope of its application.

Further work will focus on the development of agent negotiation and collaboration protocols which address the specific needs of MAS. Also the proposed approach will be validated with more complex real world scenarios. Early test indicate that the definition of appropriate negotiation and collaboration protocols in conjunction with internal agent methods will quickly result in close to optimal solutions.

ACKNOWLEDGMENT

The reported work is partially funded by the European Union as part of the NMP-2-CT-2004-507978 EUPAS project. The support is gratefully acknowledged.

REFERENCES


