Functional Modelling in Evolvable Assembly Systems

David Sanderson[1](✉), Jack C. Chaplin[2], and Svetan Ratchev[2]

[1] Centre for Aerospace Manufacturing, University of Nottingham, Easter Park, Nottingham NG7 2PX, UK
david.sanderson@nottingham.ac.uk

[2] Institute for Advanced Manufacturing, Advanced Manufacturing Building, Jubilee Campus, University of Nottingham, Nottingham NG7 2GX, UK
{jack.chaplin, svetan.ratchev}@nottingham.ac.uk

Abstract. The design and reconfiguration of adaptive production systems is a key driver in modern advanced manufacturing. We summarise the use of an approach from the field of functional modelling to capture the function, behaviour, and structure of a system. This model is an integral part of the Evolvable Assembly Systems architecture, allowing the system to adapt its behaviour in response to changing product requirements. The integrated approach is illustrated with an example taken from a real EAS instantiation.

Keywords: Architecture · Evolvable assembly systems · Functional modelling Multi-agent systems

1 Introduction

The manufacturing industry as a whole is facing increased market unpredictability and labour costs, as well as growing consumer demand for highly personalised goods and services with a shorter time to market and increased product diversity [1]. In order to incorporate these changes, manufacturing systems have begun to take advantage of adaptive control for flexibility, resilience, and monitoring. Manufacturing companies in many sectors are therefore investigating smart, flexible, and adaptive manufacturing lines that can autonomously self-heal, self-adapt, and reconfigure in response to changing product requirements. This is typified by the 'batch-size-of-one' problem, wherein each product may be unique and the manufacturing system must be capable of carrying out different production processes as required by the current product.

A common approach to these problems is that of cyber-physical systems [2], often implemented as a multi-agent system [3]. One such implementation is that of the Evolvable Assembly Systems project [4] (EAS). The EAS project addresses the challenges of design and modelling of such systems with a behavioural framework based on functional modelling. This is embedded in a multi-agent cyber-physical systems architecture, allowing the system structure and behaviour to be related to intended system function, drawing on the Function-Behaviour-Structure (FBS) formalisms of Gero, Rosenman, Umeda, and others [5–11]. This provides the capability for the system to identify when the requirement to reconfigure is triggered, and then "design the change" that implements the reconfiguration.

2 Functional Modelling and the Behavioural Approach

The modelling, design, and integration of assembly systems is primarily based on process requirements. This requires that the capabilities of a production resource are captured and can be reasoned about. There are two main ways of capturing the capabilities of a production resource: by considering the processes that the resource can perform, and by considering how their structure and behaviour relate to their intended functions. We take the second approach, based on the Function-Behaviour-Structure (FBS) formalisations developed by Gero, Rosenman, Umeda, and others in the field of functional modelling [5–11] that separate the objective structure of a system from its subjective function, and the different behaviours related to each of those. The general behavioural approach derived from functional modelling is shown in Fig. 1.

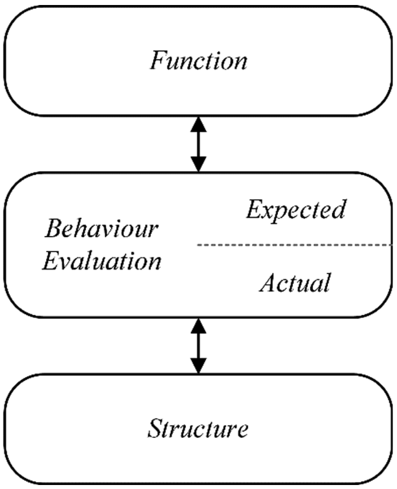

Fig. 1. Overview of the functional modelling behavioural approach

We therefore define Function as an abstraction of behaviour for a specific use or purpose, Behaviour as state transitions from input to output, and Structure as the physical model of the system and subsystem, and the connections between them. Describing a system in such a function-based manner allows for a decomposition to be performed for each of the various uses required at any given time. Applying this to modular and reconfigurable manufacturing systems requires the decomposition to be guided by the available modules [12, 13]. With this in mind, we can define Function, Behaviour, and Structure in the context of an EAS.

In any assembly system, the system **Function** is primarily to assemble a given product. Evolvable Assembly Systems however, are flexible, reconfigurable, self-adaptive systems. This leads to a set of functions to enable that adaptivity. In any manufacturing context, there is also always a set of requirements for quality. We can therefore say that the Functions of an EAS are divided into those relating to the *product requirements*, those relating to the *adaptation requirements*, and those relating to *quality requirements*.

When discussing **Behaviour**, the most important concept is to distinguish between the *expected* behaviour required by the system Function, and the *structural* behaviour expressed by the system Structure. For each of these, there are many kinds of behaviour in the system. The behaviour type covered by previous work on FBS models could be described as *non-adaptive behaviour*. This describes the steady-state production processes carried out by the assembly system, and is often a simple sequence of actions. As we are dealing with an adaptive system, we must account both for behaviours that are *adaptive processes*, and for those that handle the *reconfiguration* of the system.

As a modular, flexible, reconfigurable manufacturing system, an EAS is a complex, context-aware, collective adaptive system of systems [14]. As such, the **Structure** of the system is composed of components, *modules,* or sub-systems, the *connections* between them, and the *state* of the system. As the structure of the system may change over time – in response to adding, removing, or re-arranging components – the system structure can be optionally divided into *configurations* from which the system can select.

The concept of system **Views** is included to formalise the requirement to describe only the aspects of a system that are relevant to the task at hand [8, 9, 15]. In existing design literature, this is included to ensure that (for example) the permittivity of light of a material is only considered if it is relevant to its purpose. In our case, we can consider the system from a variety of Views, for example focussing on process capabilities when checking for manufacturability, or focussing on the topology and connections of the system where layout is important. This allows the framework to be applied at the correct level of abstraction and complexity for each specific application.

3 Evolvable Assembly Systems

The behavioural framework described in Sect. 2 is at the core of the Evolvable Assembly Systems approach. The required set of functions is used to determine the behaviours that are expected to fulfil those functions. The structure of the system is then designed, and the actual behaviour expressed by that structure is compared to the expected behaviour. The algorithmic basis for this process of distributed behavioural evaluation is described in more detail in [16].

Each resource in the system consists of some structure and corresponding behaviour(s). Some resources also have multiple configurations. These are managed by an intelligent agent; all agents in the system communicate with each other to provide distributed control based on a joint system model that provides both operational data, and the coordination for the distributed behavioural evaluation.

3.1 Multi-agent Architecture

An agent-oriented view of the EAS architecture is shown in Fig. 2. Each resource in the system is controlled at a high level by an intelligent agent – an encapsulated piece of software that makes control decisions based on available information. EAS uses the Beliefs-Desires-Intentions agent paradigm [18], so this information is stored as beliefs, and translated into immediate intentions (plans) in order to accomplish long-term desires (goals). The intentions of the agent are executed through behaviour in the system, based on the behaviours that can be expressed by the structure of the resource that is being controlled.

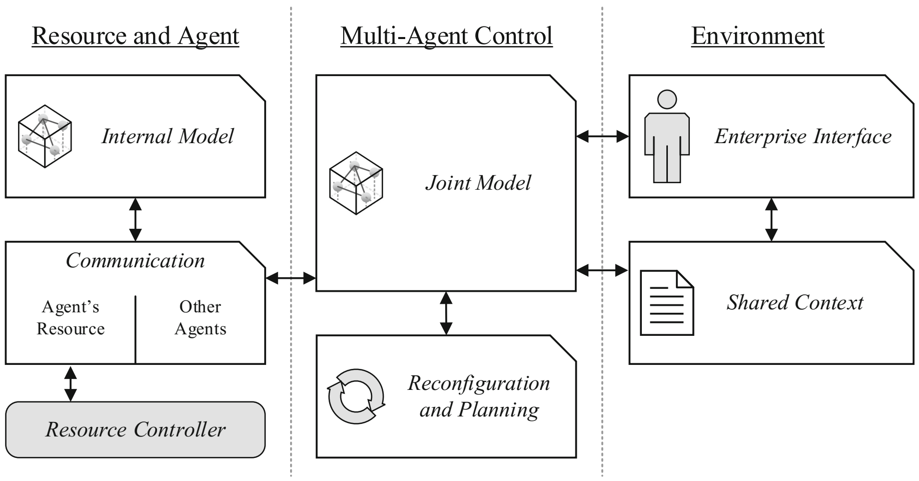

Fig. 2. An agent-oriented view of EAS architecture

Each resource in the system can be defined as a set of structures and associated behaviours using Business to Manufacturing Markup Language (B2MML) based on the ISA-95 standard [17, 18]. To enable an agent to control a resource, this description is used to generate an interface. This interface connects the agent with the PLC, controller, or similar that provides the low-level control for the resource hardware. This interface layer allows the agent core to remain the same whilst still providing control for a variety of hardware archetypes.

3.2 Shared Context

Although each agent is responsible for local control of the resource it is attached to, the collection of agents provide a distributed agent control layer for the whole production system. Communication between these agents is based on the ubiquitous sharing of contextual information.

This contextual information covers all aspects of the system. At the most fundamental level, it describes the product that is to be produced by the system, and the production capabilities of each resource in the system, as well as any "joint capabilities" resulting from combinations of resources. This is further enhanced by information about the state of the product and system, for example the pose of a robot arm, the location of a pallet, or relevant metrology data. The context also provides a link between the EAS control system and the rest of the enterprise in which it is situated. This may include control systems for other assembly cells, or higher-level enterprise information systems. All of this information is stored in a context layer that is accessible to all agents as required. This shared context is implemented using a publish-subscribe data distribution service [19].

4 Implementation Example

This system has been implemented in a real precision assembly demonstrator (PAD, shown in Fig. 3) at the University of Nottingham. The demonstrator is designed to assemble interior hinges from the automotive industry. Each product is defined by a recipe file that indicates the détente force – achieved by the configuration of ball-spring pairs added to the hinge. Because each hinge produced by the system could be unique, these recipe files are a way to formalise the batch-size of one requirements in the system.

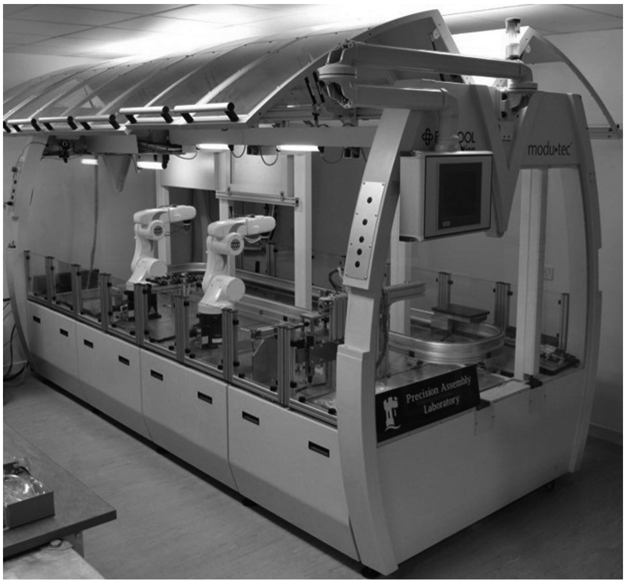

Fig. 3. The precision assembly demonstrator (PAD)

The demonstrator consists of two KUKA KR5 sixx R650 robots alongside a testing station with visual inspection and force testing equipment, connected via a linear shuttle system. Each robot has an individual working area, as does the testing station. The robots share a tool rack, giving them each access to a variety of different end effectors via an automatic tool changer. The whole system is accessible via a part loading station, where pallets of unassembled parts are loaded, and pallets with completed products are removed. A diagram of the demonstrator layout can be seen in Fig. 4; the physical connections between the production modules are shown in Table 1.

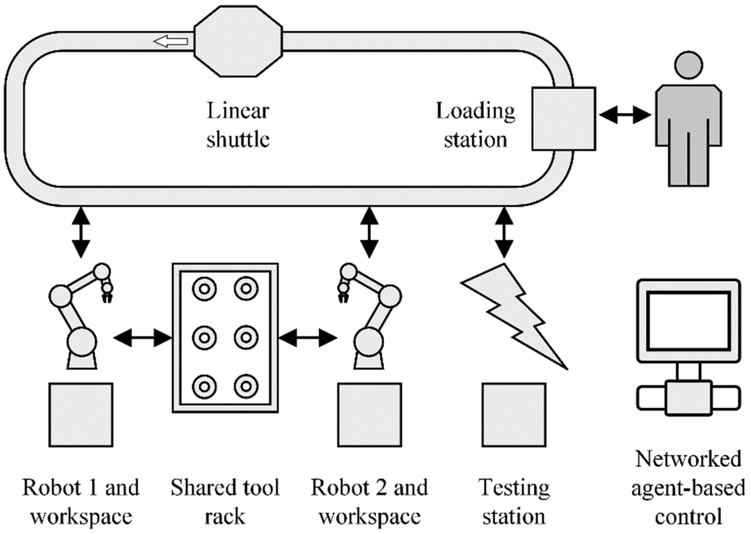

Fig. 4. Layout of the PAD

Table 1. Production modules of the PAD and their physical connections

	Linear shuttle	Loading station	Robot 1	Robot 1 Workspace	Robot 2	Robot 2 workspace	Tool rack	Testing station	Testing station workspace
Linear shuttle	▨	▨	▨		▨			▨	
Loading station	▨	▨							
Robot 1	▨		▨	▨			▨		
Robot 1 Workspace			▨	▨					
Robot 2	▨				▨	▨	▨		
Robot 2 workspace					▨	▨			
Tool rack			▨		▨		▨		
Testing station	▨							▨	▨
Testing station workspace								▨	▨

In order to assemble the hinges, each subsystem has a number of behaviours as described in Table 2. The robots have the following pick and place behaviours: move the pallet between the shuttle system and the workspace; change end effector; match the hinge leaves; insert the hinge pin; insert a variable number of springs; insert a variable number of balls; insert the cap. The testing station has pick and place behaviours to move the product between the shuttle system and the workspace, as well as a force testing behaviour to test the détente force, and a vision testing behaviour to ensure the product is correctly assembled. The shuttle system has only one behaviour: moving from one station to another. The loading station and operator can only load parts into the system and remove parts from the system.

These behaviours are then utilised by the agent controlling the relevant resource in order to produce the product specified by the recipe file – that is, accomplish the required function inherent in the new product. Each agent may further be able to adapt some or all of the structure of its resource. In the case of our example, the shared tool rack provides a selection of end effectors. The selection of a new end effector changes both the structure and behaviour of the resource.

As the "loading station" behaviours are carried out by an operator, they are logically part of the system but do not have to be programmed into an agent – the operator only needs some way to tell the rest of the system what behaviour has been carried out.

Table 2. Behaviours available to each production module in the PAD

Module	Behaviour	Sub-behaviour (optional)	Behaviour type
Linear shuttle	Move	To [*location*]	Adaptive
Loading station	Load pallet		Non-adaptive
	Unload pallet		Non-adaptive
Robot 1 & 2	Move pallet	Shuttle → workspace	Non-adaptive
		Workspace → shuttle	Non-adaptive
	Change end effector	[*Various end effectors*]	Reconfiguration
	Match hinge leaves		Adaptive
	Insert hinge pin		Adaptive
	Insert springs	[*Variable number*]	Adaptive
	Insert balls	[*Variable number*]	Adaptive
	Insert cap		Adaptive
Tool rack	Provide end effector		Reconfiguration
	Receive end effector		Reconfiguration
Testing station	Move pallet	Shuttle → workspace	Non-adaptive
		Workspace → shuttle	Non-adaptive
	Force test	[*Variable force*]	Adaptive
	Vision test		Adaptive

In addition to the behaviours available to each physical production module, there are a number of behaviours available to the distributed agent control layer. These behaviours are not linked to any specific production resource; they allow the system to adapt and reconfigure its own function, behaviour, and structure from a top-level.

One example of such a behaviour is reconfiguration of the control flow in response to the removal of a production module. Although the system is designed to allow the work to be shared between both robots, as the robots are identical and share a tool rack, the work can be done by a single robot. In this way, the system can select between the single robot configuration and the dual robot configuration and re-plan the sequence of production behaviours accordingly.

5 Summary and Acknowledgements

This paper has described the agent-oriented architecture of Evolvable Assembly Systems in the context of the FBS approach from function modelling. This approach allows the system to accurately model the adaptive structure and behaviour of the system, and leverage them to achieve the system functions required by the changing product requirements of a batch-size of one scenario. This approach has been demonstrated on a real demonstration cell at the University of Nottingham.

The authors gratefully acknowledge the support provided by UK EPSRC Evolvable Assembly Systems (EP/K018205/1).

References

1. Kagermann, H., Helbig, J., Hellinger, A., Wahlster, W.: Recommendations for implementing the strategic initiative INDUSTRIE 4.0: securing the future of German manufacturing industry, Final Report of the Industrie 4.0 Working Group (2013)
2. Monostori, L.: Cyber-physical production Systems: roots, expectations and R&D challenges. Procedia CIRP **17**, 9–13 (2014)
3. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10**, 115–152 (1995)
4. Chaplin, J.C., et al.: Evolvable assembly systems: a distributed architecture for intelligent manufacturing. IFAC-PapersOnLine **48**, 2065–2070 (2015)
5. Gero, J.S.: Design prototypes: a knowledge representation schema for design. AI Mag. **11**, 26–36 (1990)
6. Gero, J.S., Kannengiesser, U.: A function–behavior–structure ontology of processes. AI EDAM Artif. Intell. Eng. Des. Anal. Manuf. **21**, 379–391 (2007)
7. Rosenman, M., Gero, J.: Purpose and function in design: from the socio-cultural to the techno-physical. Des. Stud. **19**, 161–186 (1998)
8. Umeda, Y., Takeda, H., Tomiyama, T., Yoshikawa, H.: Function, behaviour, and structure. In: Applications of Artificial Intelligence in Engineering V, pp. 177–193 (1990)
9. Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., Tomiyama, T.: Supporting conceptual design based on the function-behavior-state modeler. Artif. Intell. Eng. Des. Anal. Manuf. **10**, 275–288 (1996)
10. Sasajima, M., Kitamura, Y.: FBRL: a function and behavior representation language. In: Proceedings of the 14th International Joint Conferences on Artificial Intelligence (IJCAI), pp. 1830–1836 (1995)
11. Mizoguchi, R., Kitamura, Y.: Foundation of knowledge systematization: role of ontological engineering. In: Roy, R. (ed.) Industrial Knowledge Management, pp. 17–36. Springer, London (2001). https://doi.org/10.1007/978-1-4471-0351-6_2

12. Koren, Y., et al.: Reconfigurable manufacturing systems. CIRP Ann. - Manuf. Technol. **48**, 527–540 (1999)
13. Koren, Y.: Reconfigurable manufacturing and beyond. In: CIRP 3rd International Conference on Reconfigurable Manufacturing (2005)
14. Sanderson, D., et al.: Advanced manufacturing: an industrial application for collective adaptive systems. In: 2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, pp. 61–67. IEEE (2015)
15. Gero, J.S., Kannengiesser, U.: Function-behaviour-structure: a model for social situated agents. In: Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions, International Joint Conference on Artificial Intelligence. pp. 101–107 (2003)
16. De Silva, L., Felli, P., Chaplin, J.C., Logan, B., Sanderson, D., Ratchev, S.: Synthesising industry-standard manufacturing process controllers. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, pp. 1811–1813. International Foundation for Autonomous Agents and Multiagent Systems (2017)
17. Business to Manufacturing Markup Language Operations Schedule Version 6.0. https://services.mesa.org/ResourceLibrary
18. ANSI/ISA-95, Enterprise-Control System Integration, Parts 1–5. https://www.isa.org/standards-publications/
19. Data Distribution Service. http://www.omg.org/spec/DDS/Current