de Silva, Lavindra and Felli, Paolo and Chaplin, Jack C. and Logan, Brian and Sanderson, David and Ratchev, Svetan (2017) Synthesising industry-standard manufacturing process controllers. In: Sixteenth International Conference on Autonomous Agents and Multiagent Systems, 8-12 May, 2017, São Paulo, Brazil.

Synthesising Industry-Standard Manufacturing Process Controllers[†]

(Demonstration)

Lavindra de Silva[*] Paolo Felli[*] Jack C. Chaplin[*]
Brian Logan[**] David Sanderson[*] Svetan Ratchev[*]
[*]Institute for Advanced Manufacturing, University of Nottingham, Nottingham, UK
[**]School of Computer Science, University of Nottingham, Nottingham, UK
first-name.last-name@nottingham.ac.uk

Keywords

Manufacturability; controller synthesis; manufacturing as a service

1. INTRODUCTION

Determining the most suitable means of producing a given product and their ordering, or *process planning* [6], is traditionally done by manufacturing engineers who are experts in the internal processes and layout of a specific factory, and, with the exception of some limited support by Computer-Aided Process Planning tools [6], is largely a manual process. From the perspective of "manufacturing as a service", where the customer's product is not known in advance, the traditional approach has drawbacks: it requires human expertise to determine whether the product can be manufactured using the resources of a given service provider, and the small batch sizes (perhaps a single item) make the manual production of process plans uneconomic. To fully realise the manufacturing as a service vision, process planning must be automated, allowing service providers to 'bid' to manufacture products in real time.

In [4], an approach was proposed to determine both *whether* a particular product is manufacturable given a set of available manufacturing resources, and *how* the product should be manufactured using those resources. In this paper, we present a tool that implements the definitions and algorithms in [4].[1] We also link the abstract representations of [4] to concrete ISA-95 standards [2], and synthesise process plans suitable for execution by industrial manufacturing systems. The core of our tool is a new reasoner for orchestrating the activities of agents in the Evolvable Assembly Systems (EAS) architecture, an agent-based architecture for manufacturing control software designed to address rapidly changing product and process requirements [3].

2. RESOURCES AND TOPOLOGY

In EAS, each *resource agent* represents and controls a manufacturing resource via a Programmable Logic Controller (PLC). A resource is either a production resource (e.g. a robot) which performs manufacturing operations on parts, or a transport resource (e.g. a conveyor or shuttle system) which moves parts between production resources. Both types of resource are modelled as labelled transition systems (LTS): states represent resource (and part) configurations, and edges represent operations that are possible at the relevant states. Special edges—called *synchronisations* and denoted IN: and OUT: —indicate the transfer of parts into and out of resources, and *nop* edges denote an "idling" operation.

We have used our tool to model the resources of the Precision Assembly Demonstrator (PAD), a real-world manufacturing system that manufactures detent hinges for the cab interiors of commercial trucks [1]. To produce a hinge, paired interior and exterior plastic leaves are attached with a metal hinge pin. Glue is applied to secure the hinge pin, and serial numbers are then engraved onto the leaves. The PAD is modelled as five resources. Resource R_1 can load a new pallet-fixture, remove a product for disposal or rework, or store it for delivery. Resource R_2 is a robotic arm that can take as input a fixture that is separated into the hinge and pin or it can take the hinge and pin as separate inputs, insert the pin into the hinge barrel, and output a single assembled part. Robot R_3 can engrave a serial number and apply glue to a given part, and R_4, the testing station, can gather and analyse image and force data. R_5 is a transport resource whose transitions represent the 'allowable' routes between production resources. Thus, while a newly loaded fixture can move from R_1 to R_2, a freshly glued part cannot move from R_3 to R_4 in order to prevent it becoming affixed to the force testing equipment. Screenshots of some LTSs modelled in our tool are shown in Fig. 1.

Together the resources form the *production topology* representing the layout of the manufacturing system. The tool computes this by taking the cross product of the LTSs representing the resources, and removing transitions with no matching 'in' and 'out' synchronisations (e.g. OUT:2 in R_5 and IN:2 in R_2). Generating the topology is exponential in the number of resources and parts in the recipe [4], but is computed only once for a given manufacturing system layout.

3. PRODUCTION RECIPES

The product to be manufactured is modelled as a *production recipe* that specifies the constituent parts, the operations required to process and assemble them into the final product, any tests that are needed to verify the product is being correctly manufactured, and how to respond to the outcome of each test, e.g., whether a product should be reworked or discarded. Recipes specify *how*, but not *where* these operations and checks should be performed. Recipes are also modelled as LTSs: edges represent *composite operations* that consume and produce parts, and states represent the state of parts in the assembly. A composite operation consists of a guard that tests some property of the product (e.g., what colour it should be, or whether two parts have been assembled correctly), followed by sequential and/or parallel compositions of primitive operations as in [4].

Fig. 1 shows a recipe for a hinge. The transition from the ini-

[†]This work was supported by the Engineering and Physical Sciences Research Council [grants EP/K018205/1 & EP/K014161/1].

[1]A video of the tool is available at https://youtu.be/SEveuikI3p8.

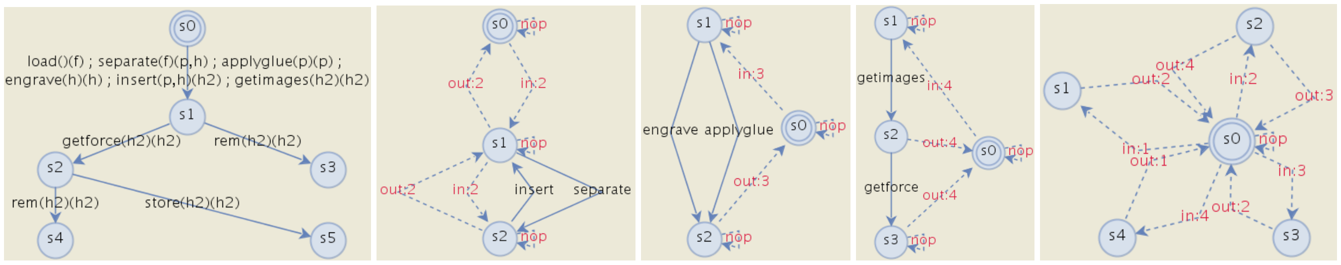

Figure 1: A recipe, and some of the simplified LTSs (R_2, R_3, R_4 and R_5) corresponding to the PAD's resources, respectively.

tial state s0 creates a new fixture *f* (without consuming any parts), separates the pin *p* and hinge-leaves *h*, applies glue to *p*, engraves a serial number on *h*, and inserts *p* into *h* to form *h2*, which is then visually examined, generating test data. Based on an analysis of the test data (not shown), *h2* is either removed from the system or hinge-force data is collected. Similarly, *h2* is either removed or sent to storage.

4. MANUFACTURABILITY & CONTROL

Our tool determines if a recipe can be manufactured on a production topology by checking if there exists a 'task simulation relation' [4] between the recipe and topology. Intuitively, task simulation associates states in the topology to states in the recipe with respect to the current allocation of parts to resources, such that each transition (composite operation) in the recipe can be executed by transitions (operations) in the topology, and the same is possible for the entire recipe, irrespective of the outcome of tests and the recipe trace that might be followed at execution-time. If a recipe is manufacturable, the tool can synthesise a *controller* that specifies how resources should be orchestrated in order to execute the recipe. Checking manufacturability and controller synthesis both involve a depth-first search of the topology to ascertain which sequence of topology transitions can execute each recipe transition.

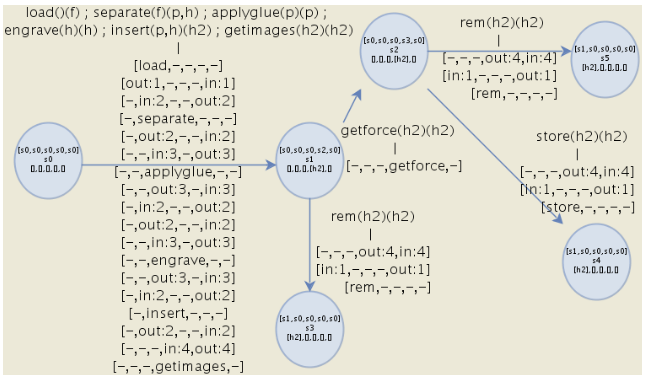

Figure 2: A controller (solution) for producing the hinge recipe.

Intuitively, a controller is an annotation of each recipe state with the corresponding resource states (i.e., topology state) and the parts being worked on in those states, and of each recipe transition with the sequence of topology transitions that need to be executed to perform the recipe transition. The screenshot in Fig. 2 shows a controller for the hinge recipe. Transition labels are displayed from top to bottom, with each row (e.g., '[load,-,-,-]') showing the operation assigned to each resource $R_1, \ldots, R_5$ (dashes represent nops).

5. INDUSTRY STANDARD CONTROL

In this section, we explain how the controller generated by our tool is converted to Business to Manufacturing Markup Language (B2MML) [2] format. B2MML implements the ANSI/ISA-95 standards in XML, and is being increasingly widely adopted in manufacturing [5]. It defines the data and process models that form the interface to manufacturing execution systems (MESs) [7] that manage and monitor the work-in-progress on the shop floor (e.g., via PLCs), and provides a convenient hardware interface for information systems, including the agents comprising the EAS.

In B2MML, each operation is specified as an *operations request* that encodes the parameters to be passed to the equipment controllers (e.g., the specific tool to be used for engraving), the material (part) requirements for each step, and any particular requirements for the resource to be used (e.g., tolerances on the operation to be performed). While some of this information must be manually entered into "operations templates" when resources are modelled as LTSs, the tool fills in the rest automatically after a controller is synthesised.

Given a controller, the tool generates its corresponding *operations schedule*, which is an AND-OR tree representing an "unfolding" of the LTS (graph) that defines the controller. This requires extracting certain details from the controller, such as: *(i)* the equipment that was assigned to perform particular operations; *(ii)* the materials consumed and produced by them; and *(iii) when* parts need to be transferred and *which* pairs of resources need to work together in order to achieve this; this 'transfer' information is inserted into the operations schedule as new operations requests. On reaching a choice point during the execution of an operations schedule, the first operations request, i.e., the test on each outgoing branch, is executed until all branches have been tried or one of the operations returns 'success'. In the latter case, the rest of that branch is executed until the next choice point is reached.

6. CONCLUSION & FUTURE WORK

We have presented a tool that implements the formalism in [4], but also synthesises B2MML controllers for real-world MESs. We plan to extend our tool (and the approach in [4]) to allow multiple resources to perform parallel operations on the *same part*, and to account for nondeterministic (uncontrollable) outcomes during production. We also plan to make topology generation more efficient, by adding the relevant outgoing topology transitions incrementally, starting from the initial states of the resources.

REFERENCES

[1] N. Antzoulatos, E. Castro, L. de Silva, A. D. Rocha, S. Ratchev, and J. Barata. A multi-agent framework for capability-based reconfiguration of industrial assembly systems. *International Journal of Production Research*

(IJPR), 2016.

[2] *Business To Manufacturing Markup Language (B2MML)*, 2015 (accessed February 11, 2017). `https://isa-95.com/b2mml`.

[3] J. C. Chaplin, O. J. Bakker, L. de Silva, D. Sanderson, E. Kelly, B. Logan, and S. M. Ratchev. Evolvable Assembly Systems: A distribted architecture for intelligent manufacturing. In A. Dolgui, J. Sasiadek, and M. Zaremba, editors, *Proceedings of the 15th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2015)*, volume 48 of *IFAC-PapersOnLine*, pages 2065–2070, Ottawa, Canada, May 2015. Elsevier.

[4] L. de Silva, P. Felli, J. C. Chaplin, B. Logan, D. Sanderson, and S. Ratchev. Realisability of production recipes. In G. A. Kaminka, M. Fox, P. Bouquet, H. E., D. F., D. V., and van Harmalen F., editors, *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI-2016)*, pages 1449–1457, The Hague, The Netherlands, August 2016. ECCAI, IOS Press.

[5] D. Emerson, H. Kawamura, and W. Matthews. Plant-to-Business (P2B) interoperability using the ISA-95 standard. *Yokogawa Technical Report*, 43:17, 2007.

[6] M. P. Groover. *Automation, production systems, and computer-integrated manufacturing*. Prentice Hall Press, 2007.

[7] B. Scholten. *MES Guide for Executives: Why and how to Select, Implement, and Maintain a Manufacturing Execution System*. International Society of Automation, 2009.