Modelling and Orchestration of Service-Based Manufacturing Systems via Skills

Julius Pfrommer*, Denis Stogl†, Kiril Aleksandrov‡, Viktor Schubert‡ and Björn Hein†

*Fraunhofer Institute of Optronics, Systems Technology and Image Exploitation (IOSB), 76131 Karlsruhe, Germany
Email: julius.pfrommer@iosb.fraunhofer.de
†Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
Email: {denis.stogl|bjoern.hein}@kit.edu
‡FZI Research Center for Information Technology, 76131 Karlsruhe, Germany
Email: {aleksandrov|schubert}@fzi.de

***Abstract*—Shortening product lifecycles and small lot sizes require manufacturing systems to adapt increasingly fast. Many existing machine tools, handling and logistics systems are already generic and not bound to a specific product a-priori. Yet this flexibility and reconfigurability on the asset level is lost in automated systems that are limited to executing a small set of predefined actions in a fixed sequence. The SkillPro[1] project aims to develop a holistic service-oriented framework for modelling and orchestration of modern adaptable manufacturing systems. The core concept is a unified abstraction for manufacturing tasks: skills provided by the available assets and the requirements of the different production steps. The skill-based system model enables the transition from generic high-level descriptions to low-level formats that can be directly executed. Self-describing assets can be added, changed and removed at runtime, taking into account technical and economic conditions to best achieve the manufacturing goals.**

I. Introduction

Service-oriented architectures in manufacturing systems have been discussed in the literature for some time [1] [2] [3]. Recently, automated service composition [4] [5] [6] has been translated into manufacturing and robotics contexts [7] [8] [9]. The SkillPro project aims to combine this approach with plug-and-produce functionality [10] [11] and the ability to generate executable processes ad-hoc from a description of the generic skills provided by available assets and the production task at hand. Therefore, SkillPro presents a new type of cyber-physical control system in extension of todays Manufacturing Execution Systems (MES), integrating data from CAD/CAM and the digital factory on one side and sensor and machine data on the other side for runtime orchestration. With this, SkillPro achieves self-adaptation of manufacturing systems to ever-changing requirements, products, and internal system states, securing execution and reconfiguration under the fulfillment of technical and economic conditions. The framework can be adapted to the requirements of manufacturing enterprises of various scale and complexity in their asset park.

[1]Skill-based Propagation of Plug'n'Produce-Devices in Reconfigurable Production Systems by AML; The project SkillPro (Grant 314247) has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no ICT-287733.

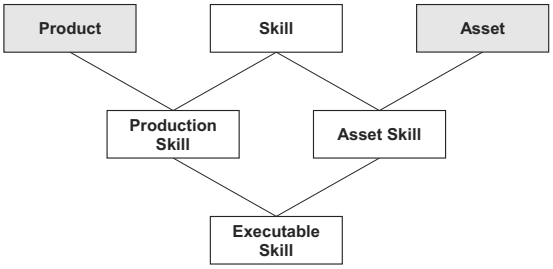

Figure 1. Skill types used in the SkillPro project.

The approach taken by SkillPro was shown to be feasibile at midterm of the project, when a working prototype of the architecture was used to control a manufacturing scenario that integrated both physical and simulated manufacturing assets. The physical assets included conveyor belts, robots, an autonomous transportation platform and dedicated safety areas with surveillance equipment. Using the approach described in the following sections, the system achieved the manufacturing goals autonomously, starting from a model of the current system state and the manufacturing goals.

II. Modelling skills in manufacturing systems

The starting point of SkillPro is a logical extension of the classical product process resource (PPR) concept in the production context. The available assets as well as the required production steps are characterized via a description format for skills from specific domains. An AutomationML-based [12] format is used for storage and communication of the skill descriptions. We now introduce the skill concepts and their relations depicted in Figure 1. Also see [13] [14] for a more in-depth discussion and the differences from prior work.

Asset: According to Haider [15] an asset is a component with economic life greater than 12 months that creates and maintains its value by providing services to users. In SkillPro the focus lies on production resources that are regarded as standalone assets (e.g. a CNC machine) or bundled together with other resources (e.g. a robot manipulator with a gripper). Therefore, assets are represented through their configurational

and functional structure. All assets are assumed to have a representation in the virtual world (e.g. as objects in an OPC-UA namespace) where they provide data and services and can request these from others. So they constitute cyber-physical objects.

Product: Products represent the input and output of a production operation, including intermediate products, material, and so on. Note that they denote product types rather than individual workpieces.

Skill: Skills refers to a type of manufacturing, logistics or other production related process (such as information exchange, retooling, etc.) and provide the structure for describing these processes in their respective domain. Skills form a hierarchy where derived skills inherit the attributes of their parent. As an example, *arc-welding* would be derived from *welding*, inheriting the *temperature* attribute.

Asset skill: An asset skill is the instantiation of a skill. It indicates the ability of a specific asset to execute the skill under the constraints described in the attributes. For example a welding robot would indicate a range for the temperature attribute specified in the *welding* skill it implements. A drilling machine could use this mechanism to describe the different materials and drill diameters it currently supports.

Production skill: A production skill gives the requirements that an asset must fulfill in order to execute a product transformation. Note that production skills state necessary requirements an asset must fulfill to be compatible. But they are not sufficient to guarantuee successful execution without additional information.

Executable skill: An executable skill is associated both with the production skill it implements and the (possibly several) relevant asset skills on the assets that are taking part in the execution. Whereas a mere matching of production skills with asset skills might come up with non-feasible assignments (whose conditions could not be captured by the skills descriptions available for that domain), the executable skills are known to be successfully executable when triggered as a service provided by the asset.

III. The SkillPro Architecture

The three main component types of the SkillPro system architecture and their relations are shown in Figure 2. They communicate (for configuration, planning, and control) via OPC-UA [16]. This allows a flexible adaptation of the data model at runtime, remote procedure calls, and push-notifications by subscription.

A. Asset Management System (AMS)

The skill-based AMS constitutes the *manufacturing-specific knowledge base* of the company. It provides the following functionality: managing assets regarding their lifecycle and their skills lifecycles; matching production and resource skills in order to configure new executable skills in collaboration with the autonomous skill execution engines (SEE) assigned to each asset; checking feasibility for a certain production order; identifying technical bottlenecks or underutilized assets; providing production alternatives by breakdown-related replanning. The AMS further gathers and processes information about the running production system to assist in asset lifecycle decisions (e.g. calculating the TCO (total cost of ownership) and lifetime value of an asset) and for interaction with customers (e.g. giving an estimation of delivery times on a long-term horizon). Based on the enterprise's size and organisation, a single AMS can be responsible for multiple production sites, managing assets and skills in order to fully maximize and capitalize on the flexibility and the reconfigurability of the asset park and the corresponding skill set.

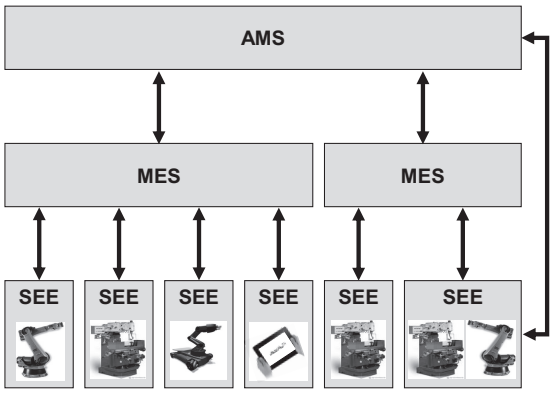

Figure 2. The SkillPro system architecture.

B. Manufacturing Execution System (MES)

The MES (usually one per production site) are responsible for the orchestration of manufacturing processes on many assets in parallel in order to achive short- to mid-term manufacturing goals. For this, the MES compute an execution plan comprised of executable skills based on a model of the manufacturing system dynamics and its current state. The MES are not directly connected to the available assets. They rather communicate with the SEE (who provide a unified interface) to retrieve state updates and to trigger executable skills by calling the appropriate services. If an unexpected event (like a machine breakdown) occurs, or if the plan horizon comes too close, the execution plan is recomputed based on up-to-date information. Lastly, the MES are not only consuming services, but also provide services where orders (the manufacturing goals) can be placed, modified and tracked by the AMS and additional user interfaces.

C. Skill Execution Engine (SEE)

In contrast to AMS and MES, the SEE are without clear precedent in automation. The central task of the SEEs is to provide smart wrappers to the physical assets (currently a single asset per SEE), which range from simple machines with little configurability to human workers. For the latter, appropriate user interfaces have been developed in the SkillPro project, each connected to an SEE-backend. This is useful to combine skill-based automated manufacturing systems with manually operated manufacturing steps (e.g. calling and

instructing service-technicians when failures occur or when maintenance/tooling procedures are scheduled). The SEE are self-aware in a sense that they contain an AML description of the wrapped asset, the current state of the asset and detailed descriptions on how to run the executable skills that can be triggered by the MES. Some SEE can communicate with each other for collaboration and synchronisation. Alternatively, time-synchronised collaboration of SEE as part of an executable skill can be triggered by the MES, making use of clock synchronisation [17].

IV. Formal Manufacturing System Dynamics

In the following, we present the formal pre- and post-conditions for the execution of executable skills. The state s of assets $a \in A$ (assuming for simplicity that assets are always operational) is the conjunction of an asset configuration $c \in C_a$ and the (amount of) currently contained products stored in a multiset ρ. Remember that product types $p \in P$ can denote also intermediay product types and material. The empty set $\varnothing$ indicates the absence of any product.

$$s = (c, \rho) \in S_a \subseteq C_a \times \mathbb{N}_0^P$$

For a mobile robot, the configuration might describe its position, whilst the configuration of a NC-mill might indicate the type of milling head currently used. The global system state is captured in a vector $\boldsymbol{\sigma}$ of time-indexed states for the assets $a \in A$.

$$\sigma_a = (s, t) \in S_a \times \mathbb{R}$$

Its semantics is the next time-index $\sigma_{a,t}$ when asset a becomes available with state $\sigma_{a,s}$. Until then, the asset executes (a series of) executable skills during which its exact state is not known.

An executable skill e is comprised of a set of participating assets $\mathcal{A}_e$, as well as pre- and post-states $s_{e,a}^{\text{pre}}, s_{e,a}^{\text{post}}$ and timing conditions $t_{e,a}^{\text{slack}}, t_{e,a}^{\text{dur}}$ for each $a \in \mathcal{A}_e$.

$$e = (\mathcal{A}_e, \boldsymbol{s}_e^{\text{pre}}, \boldsymbol{s}_e^{\text{post}}, \boldsymbol{t}_e^{\text{slack}}, \boldsymbol{t}_e^{\text{dur}})$$

The time slack $t_{e,a}^{\text{slack}}$ gives the amount of time—after starting the execution of e—until a joins in the execution. The duration of e on asset a (after a possible slack time) is $t_{e,a}^{\text{dur}}$. Given an initial system state $\boldsymbol{\sigma}$, the execution of some e results in a new system state $\boldsymbol{\sigma}^{(e)}$, indicating the time and state where assets $a \in A$ next become available. A sequence of executable skills can be written as $w = e_1 e_2 \ldots$. The single rule that governs the transition between any $\boldsymbol{\sigma}^{(w)}$ and $\boldsymbol{\sigma}^{(we)}$ consists of the following pre- and post-condition:

$$\frac{\forall a \in \mathcal{A}_e,\ \sigma_{a,s}^{(w)} = s_{e,a}^{\text{pre}}}{\begin{aligned} t^{\text{start}} &= \max\left\{t^{\text{now}} \cup_{a \in \mathcal{A}_e} \left(\sigma_{a,t}^{(w)} - t_{e,a}^{slack}\right)\right\} \\ \sigma_a^{(we)} &= \begin{cases} \left(s_{e,a}^{\text{post}}, t^{\text{start}} + t_{e,a}^{\text{slack}} + t_{e,a}^{\text{dur}}\right), & \text{if } a \in \mathcal{A}_e \\ \sigma_a^{(w)}, & \text{else} \end{cases} \end{aligned}}$$

If the pre-condition is fulfilled, then e can be executed to reach the state $\boldsymbol{\sigma}^{(we)}$ stated by the post-condition. Note that the above transition rule can be used to describe the system dynamics of a manufacturing system, including a) the transformation of products, b) transportation and storage, c) concurrency, i.e. parallel execution on many assets and synchronization for collaboration, and d) changes to resource configurations (e.g. tooling, or changing positions of a mobile robot) that might occur as part of a manufacturing task or as a dedicated procedure. To allow for more sophisticated reasoning, higher-level abstractions can be constructed, using this definition of executable skills as the basic building block.

V. Plug-and-Produce and Mass-Customization via Runtime Configuration and Orchestration

When an asset, represented by a SEE, is first introduced within the system, it sends its AutomationML description (including the provided asset skills) to the AMS. After that a configuration takes place, where the asset is positioned is topologically and logically connected to the neighboring assets. A virtual representation of the SEE is created in the OPC-UA address space of the runtime system, containing all the data from asset's description as well as the services for interacting with the SEE. Then the available knowledge about the assets is combined with information about specific products to generate and run automated manufacturing procedures. Since new products can also be added at runtime, the SkillPro approach enables mass-customization settings where small lot-sizes of bespoke products are manufactured in an automated system.

A. Assigning assets to production steps

The AMS manages the skill-based bill of processes for a given product. The possible production steps (product transformations) are known a-priori and form a relation on the multiset of product types $\tau \in \mathbb{N}_0^P \times \mathbb{N}_0^P$, indicating which and how many (intermediary) input products of type $p \in P$ are transformed into which output products. The product transformations also indicate which production skills are needed to accomplish them. Valid assignment can thus be stated as results of a constraint network on a set of variables linking the attributes of possible configurations of assets and attribute requirements of the production skills related to the transformation τ. The AMS finds the possible set of assets and their configurations for the different production steps using an Adaptive Genetic Algorithm.

B. Generating executable skills

The executable skills are generated and confirmed in collaboration between the AMS and the SEEs. This step will usually be assisted by human operators for non-trivial skill domains. Depending on how well the skill can be understood from a-priori available information, the implementation of the executable skills on the SEE may stem from one of the following sources:

- Use of a custom procedure for the executable skill that can be individually programmed asset-dependent code (e.g. a PLC function block) or asset-independent code from a central repository (e.g. a robot program)

- Use of a pre-loaded, generic procedure with a set of input parameters (e.g. navigation of a mobile platform)
- Auto-generation of procedures from a high-level task description. Several such formats have been developed for specific domains, such as onology-based assembly information [18] [19].

C. Computing and running the execution plan

For every order, a high-level production plan made up of executable skills is generated by the AMS. It contains (possibly several) sequences of production steps for a particular product. Various economic parameters are already taken into consideration at this stage, as the AMS tracks the orders on a long-term horizon, the planned availability of assets, and performance indicators such as OEE (overall equipment effectiveness) and TCO. However this coarse production plan does not suggest a precise short-term time schedule and does not take the runtime state of the manufacturing system into consideration.

The MES receives the orders with the high-level production plan from the AMS. It then needs to find a (time-indexed) execution plan to transition the current system state into one that achieves the goal description (e.g. where all ordered products are in stock). This appraoch is inspired by McCarthy's original Situation Calculus [20]. The model of the system dynamics (the available executable skills) and the current system state are fed into a planner (based on the Fast-Downward algorithm [21]) to derive a sequence of executable skills that reaches the predefined goals. In the future, additional information (such as material costs or energy consumption) attached to the executable skills will be used for multi-objective decision making in the planning process.

This execution plan is then fed into the control loop of the MES and the executable skills are triggered on the SEE accordingly by calling a remote procedure via OPC-UA. The same communication channel is used to keep the MES knowledge up to date about the SEE's state. Unexpected events (like a time delay or quality problems) can mostly be handled by a replanning step that takes the updated system state into account. Otherwise the AMS is triggered and can notify the human operators. This mechanism is also used to accomplish planned reconfigurations of assets with human assistance.

VI. Summary and Outlook

The skill-based service-oriented manufacturing framework developed in the SkillPro project enables production systems to adapt automatically to changing system states and external requirements. In this work, we presented the overall system design and its rationale. Furthermore, we showed how high-level information about the manufacturing system and the products is used at runtime to derive automated manufacturing processes for execution. An emphasis was made to provide clean abstractions and interfaces for users and standardisation bodies to enhance the framework with AutomationML-based models for skill descriptions and reasoning facilities for their specialised domains. The SkillPro consortium aims to release the main architecture components as open-source, relying on appropriate tools for OPC-UA and AutomationML. Thus, future work can rely on a proven infrastructure.

References

[1] F. Jammes and H. Smit, "Service-oriented paradigms in industrial automation," *Industrial Informatics, IEEE Transactions on*, vol. 1, no. 1, pp. 62–70, 2005.

[2] L. M. S. De Souza, P. Spiess, D. Guinard, M. Köhler, S. Karnouskos, and D. Savio, "Socrades: A web service based shop floor integration infrastructure," in *The Internet of Things*. Springer, 2008, pp. 50–67.

[3] G. Cândido, A. W. Colombo, J. Barata, and F. Jammes, "Service-oriented infrastructure to support the deployment of evolvable production systems," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 4, pp. 759–767, 2011.

[4] D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella, "Automatic composition of e-services that export their behavior," in *International Conference on Service-Oriented Computing (ICSOC)*. Springer, 2003, pp. 43–58.

[5] P. Traverso and M. Pistore, "Automated composition of semantic web services into executable processes," in *The Semantic Web — ISWC 2004*. Springer, 2004, pp. 380–394.

[6] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, "Htn planning for web service composition using shop2," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, no. 4, pp. 377–396, 2004.

[7] F. Tao, D. Zhao, Y. Hu, and Z. Zhou, "Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system," *Industrial Informatics, IEEE Transactions on*, vol. 4, no. 4, pp. 315–327, 2008.

[8] M. Loskyll, J. Schlick, S. Hodek, L. Ollinger, T. Gerber, and B. Pirvu, "Semantic service discovery and orchestration for manufacturing processes," in *Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2011, pp. 1–8.

[9] D. Di Marco, P. Levi, R. Janssen, R. van de Molengraft, and A. Perzylo, "A deliberation layer for instantiating robot execution plans from abstract task descriptions," in *Proceedings of the International Conference on Automated Planning and Scheduling: Workshop on Planning and Robotics (PlanRob)*. AAAI Press, 2013.

[10] T. Arai, Y. Aiyama, Y. Maeda, M. Sugi, and J. Ota, "Agile assembly system by "plug and produce"," *CIRP Annals-Manufacturing Technology*, vol. 49, no. 1, pp. 1–4, 2000.

[11] M. Onori, N. Lohse, J. Barata, and C. Hanisch, "The IDEAS project: plug & produce at shop-floor level," *Assembly automation*, vol. 32, no. 2, pp. 124–134, 2012.

[12] R. Drath, A. Lüder, J. Peschke, and L. Hundt, "AutomationML-the glue for seamless automation engineering," in *Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2008, pp. 616–623.

[13] J. Pfrommer, M. Schleipen, and J. Beyerer, "PPRS: Production skills and their relation to product, process, and resource," in *Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2013, pp. 1–4.

[14] K. Aleksandrov, V. Schubert, and J. Ovtcharnova, "Skill-based Asset Management: A PLM-approach for Reconfigurable Production System," in *Proceedings of the 11th IFIP WG5.1 International Conference on Product Lifecycle Management*. Springer, 2014.

[15] A. Haider, *Information Systems for Engineering and Infrastructure Asset Management*. Springer, 2013.

[16] S.-H. Leitner and W. Mahnke, "OPC UA–service-oriented architecture for industrial applications," *ABB Corporate Research Center*, 2006.

[17] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 147–163, 2002.

[18] K.-Y. Kim, D. G. Manley, and H. Yang, "Ontology-based assembly design and information sharing for collaborative product development," *Computer-Aided Design*, vol. 38, no. 12, pp. 1233–1250, 2006.

[19] L. Da Xu, C. Wang, Z. Bi, and J. Yu, "AutoAssem: an automated assembly planning system for complex products," *Industrial Informatics, IEEE Transactions on*, vol. 8, no. 3, pp. 669–678, 2012.

[20] R. Reiter, *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT press, 2001.

[21] M. Helmert, "The Fast Downward Planning System." *J. Artif. Intell. Res.(JAIR)*, vol. 26, pp. 191–246, 2006.