A formal skill model to enable reconfigurable assembly systems

Lauren Van De Ginste a,b, Alexander De Cock c, Axl Van Alboom a,b, Stijn Huysentruyt a,b, El-Houssaine Aghezzaf a,b and Johannes Cottyn a,b

a Department of Industrial Systems Engineering and Product Design, Ghent University, Technologiepark 46, 9052 Gent-Zwijnaarde, Belgium; b Industrial Systems Engineering (ISyE), Flanders Make vzw, 8500 Kortrijk, Belgium; c CodesignS, Flanders Make vzw, 3920 Lommel, Belgium

ARTICLE HISTORY
Compiled August 26, 2022

ABSTRACT
As assembly systems move into the era of mass customisation, the complexity of design processes, (re)configurations and operations rises. Well-structured data are key in keeping this complexity manageable. Hereto, this paper presents a multidimensional formal skill model designed to deliver generic descriptions of needs and capacities with skills as the connector between products, processes and resources. The model formalises resource structures in relation to the processes they master and products they can produce. This paper discusses the case-based evaluation in a reconfigurable assembly system and highlights the added-value of a skill-based modelling approach. The presented formal model combines concepts coming from both offline and online modelling perspectives and allows for various applications and levels of detail. The resource structures embedded in the prerequisites of a skill enable matchmaking of resources for workspace design and reconfigurations. The mapping of the model to standardised ISA-95 models couples production needs to the resources allowing for more optimal production planning, control and a structured interface between enterprise and control systems. The possibility to couple states to the assembly environment allows for optimal runtime orchestration.

KEYWORDS
flexible assembly; mass customisation; reconfigurable assembly system; skill-based engineering; formal modelling

1. Introduction

In the history of the industrial age, technological and socio-economical changes have frequently altered customer demand (Antzoulatos et al. (2017)). In recent years, companies have been faced with customers who are no longer passive clients. They push companies to more active involvement and they ask for personalised, customised products closer to their needs and desires (Pollard, Chuo, and Lee (2008); Kucukkoc and Zhang (2017)).

Mass customisation has mainly an impact on manufacturing companies and their assembly processes (Salunkhe and Fast-Berglund (2020)). If companies want to remain competitive, their assembly systems should be agile and reconfigurable (Battaia et al. (2018)). Companies have to efficiently adapt to the required changes in
processing functions, production capacity, and dispatching of orders by rapidly adaptable resources (Hoang and Fay (2019); Järvenpää et al. (2019)). New Industry 4.0 (I4.0) enabled technologies can help companies to face these mass customisation challenges and offer the opportunity to create new, more dynamic and flexible assembly configurations (Doigni, Sgarbossa, and Simonetto (2021)). However, it is recognised that data-driven models and related methods to design, configure, control and maintain I4.0 assembly systems are indispensable. There is an urgent need for generic methodologies representing the enormous diversity and possibilities in configurations of machines, tools and auxiliary devices used to produce or assemble the products (Vichare et al. (2009)).

Machine interpretable skill-based models with skills as a connector between the products, processes and resources could represent this enormous diversity and facilitate production planning and reconfiguration processes (Backhaus and Reinhart (2017); Hoang and Fay (2019)). They could allow for rapid system design in heterogeneous multi-vendor production environments (Järvenpää et al. (2019)) and are seen as key for task-oriented programming of complex robots (Rovida and Krüger (2015)). The potential and importance of formal models and ontologies to capture and share company-specific assembly knowledge cannot be neglected in the collaboration across various domains from assembly design to process planning (Imran and Young (2015)).

In this context, a conceptual formal skill model was put forward (Van De Ginste et al. (2021)). The development process and implementation are discussed and the model is evaluated on a Reconfigurable Assembly System (RAS). A RAS comprises the characteristics of a Reconfigurable Manufacturing System (RMS) with relevance to assembly (Jefferson et al. (2013)). RMSs and RASs have been frequently proposed to meet the changes and uncertainties of manufacturing environments by reconfiguring hardware and/or software resources (Bi et al. (2008)). The model combines concepts coming from both offline and online modelling perspectives. This way the model can help the decision support on hierarchical levels in assembly systems to answer questions as 'what will (or can) happen?' and 'how should it happen?' (Jaskó et al. (2020)).

We pursue the main contribution of this paper answering the following three research questions: (1) What elements form the basis of a formal skill model to facilitate decision support on hierarchical levels in assembly?; (2) How should a formal skill model for assembly systems look like?; (3) How well does this model behave in practice on a reconfigurable assembly system?

The paper is organised as follows. Section 2 analyses current production research on skill-based modelling. It answers the first research question by extracting the common ground and required fundaments for a skill model found in literature. Section 3 describes the skill model and deepens the fundaments. Section 4 dives into the case-specific instantiation of the model for a RAS and describes a case-based evaluation of the formal skill model. Section 5 discusses the use, key strengths and drawbacks of the skill-based modelling approach for a RAS. Finally, conclusions and further research directions are found in section 6.

2. Related work

To deal with the growing complexity in the manufacturing and assembly domain, several research projects have looked into skill-based (or capability-based) modelling. Most models form an independent link between processes, products and resources via skills (Backhaus and Reinhart (2017)).
However, different levels of detail, methods and purposes arise (Järvenpää et al. (2019); Hoang, Hildebrandt, and Fay (2018)). Especially on the meaning and use of the term ‘skill’ versus ‘capability’, multiple viewpoints exist. Previous work showed that, as the here-introduced model is broad and de facto spans both terms, the term ‘skill’ is a logical choice for our research (Van De Ginste et al. (2021)). The State-Of-The-Art (SOTA) in skill-based modelling is discussed in section 2.1. The pitfalls as well as the fundaments for a formal skill model are summarised in section 2.2.

2.1. Skill-based models

Two categories of modelling perspectives, each with their specific sub-applications, can be identified in research on skill-based models (Köcher et al. (2020b)). The first perspective focuses on (semi-)offline tasks during design and reconfigurations. Such models mostly provide a definition of concepts and relations, as well as properties. The second perspective mostly aims at online interventions during execution encapsulating machine functionalities. Only a limited amount of models can be seen as multi-perspective models embedding both perspectives. Table 1 gives an overview of the various applications of SOTA models in both categories.

The numerous views on the skill concept are not only noticeable in the inconsistent use of terminologies (capability versus skill) and various applications, but also in the specific interpretation of the skill concept in each model. Malakuti et al. (2018) recognise that the skill concept is in fact four dimensional. Skills can be atomic or composed, product-independent or product-specific, process-independent or process-specific and even resource-independent or resource-specific. A mapping of the four dimensions on the analysed models is presented in table 2. The broader the interpretation of the skill concept, the more applications fit in. Various SOTA models are discussed per category in the following three subsections.

2.1.1. (Semi-)offline modelling perspective

The first half of the models presented in table 1 target mainly (semi-)offline applications and consist of formal models and ontologies to create skill structures that can be used as a shared vocabulary or for optimisation and reasoning purposes. Agyapong-Kodua, Haraszkó, and Németh (2014) for example defined a ‘recipe-based’ approach to design a factory facility based on the semantic skill modelling of resources and matching with product-process requirements. Recently, ontologies using Web Ontology Language (OWL) have also been considered to capture information of machines and their skills. Weser et al. (2020) introduced an ontology with realisation-neutral skills for multi-layered feasibility checking, which evaluates the compatibility of resources and their offered skills with the requirements of manufacturing tasks at symbolic and sub-symbolic levels.

Järvenpää et al. (2019) developed a model to support automatic matchmaking between product requirements and resource capacities, allowing both atomic and combined skills. Later on, their model was extended towards the online plug-and-produce concept via executable skills for task-oriented programming in modular production systems (Siltala, Järvenpää, and Lanz (2018)). Because their executable skills are not originally embedded in their skill descriptions, an additional eXtendable Markup Language (XML) description is used which results in no direct link between their more abstract and executable skills. Similarly, Antzoulatos et al. (2017) presented a multi-agent system developed on the Java Agent DEvelopment (JADE) platform that can
### Table 1. Skill-based modelling: state-of-the-art on applications

<table>
<thead>
<tr>
<th>Research contribution</th>
<th>Perspective&lt;sup&gt;a&lt;/sup&gt;</th>
<th>Application&lt;sup&gt;a&lt;/sup&gt;</th>
<th>Matchmaking</th>
<th>Production planning</th>
<th>Runtime orchestration&lt;sup&gt;b&lt;/sup&gt;</th>
<th>Task-oriented programming</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Semi-) Offline</td>
<td>Online</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Agyapong-Kodua, Haraszkó, and Németh (2014)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Weser et al. (2020)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Antzoulatos et al. (2017)</td>
<td>•</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Perzylo et al. (2019)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Bennulf et al. (2021)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Hashemi-Petroodi et al. (2019)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Backhaus and Reinhart (2017)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. Danny et al. (2017)</td>
<td>0</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. Profanter et al. (2017)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. Pedersen et al. (2016)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15. Schou et al. (2018)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. Köcher et al. (2020b) &amp; Köcher et al. (2020a)</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<sup>a</sup> • = initial scope, ◦ = extended scope; <sup>b</sup> E.g. plug-and-produce, autonomous set-up, operator support and disturbance handling

### Table 2. Skill-based modelling: interpretation of skill concept

<table>
<thead>
<tr>
<th>Research contribution</th>
<th>Perspective&lt;sup&gt;a&lt;/sup&gt;</th>
<th>Skill concept&lt;sup&gt;b&lt;/sup&gt;</th>
<th>Composition of skills</th>
<th>Resource dependency</th>
<th>Process dependency</th>
<th>Product dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Semi-) Offline</td>
<td>Online</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Agyapong-Kodua, Haraszkó, and Németh (2014)</td>
<td>•</td>
<td></td>
<td>Yes</td>
<td>⊕</td>
<td>⊖</td>
<td>⊖</td>
</tr>
<tr>
<td>2. Weser et al. (2020)</td>
<td>•</td>
<td></td>
<td>Yes</td>
<td>⊥</td>
<td>⊕</td>
<td>⊖</td>
</tr>
<tr>
<td>4. Antzoulatos et al. (2017)</td>
<td>•</td>
<td>0</td>
<td>Yes</td>
<td>⊗</td>
<td>⊖</td>
<td>⊖</td>
</tr>
<tr>
<td>5. Perzylo et al. (2019)</td>
<td>•</td>
<td></td>
<td>Yes</td>
<td>⊗</td>
<td>⊖</td>
<td>⊖</td>
</tr>
<tr>
<td>6. Bennulf et al. (2021)</td>
<td>•</td>
<td></td>
<td>Yes</td>
<td>⊗</td>
<td>⊖</td>
<td>⊖</td>
</tr>
<tr>
<td>7. Hashemi-Petroodi et al. (2019)</td>
<td>•</td>
<td></td>
<td>Yes</td>
<td>⊗</td>
<td>⊖</td>
<td>⊖</td>
</tr>
<tr>
<td>10. Backhaus and Reinhart (2017)</td>
<td>•</td>
<td></td>
<td>Yes</td>
<td>⊗</td>
<td>⊖</td>
<td>⊖</td>
</tr>
<tr>
<td>11. Danny et al. (2017)</td>
<td>0</td>
<td>•</td>
<td>Yes</td>
<td>⊥</td>
<td>⊕</td>
<td>⊖</td>
</tr>
<tr>
<td>12. Profanter et al. (2017)</td>
<td>•</td>
<td></td>
<td>Yes</td>
<td>⊥</td>
<td>⊕</td>
<td>⊖</td>
</tr>
<tr>
<td>13. Rovida and Krüger (2015)</td>
<td>•</td>
<td></td>
<td>No</td>
<td>⊕</td>
<td>⊖</td>
<td>⊖</td>
</tr>
<tr>
<td>14. Pedersen et al. (2016)</td>
<td>•</td>
<td></td>
<td>No</td>
<td>⊕</td>
<td>⊖</td>
<td>⊖</td>
</tr>
<tr>
<td>15. Schou et al. (2018)</td>
<td>•</td>
<td></td>
<td>No</td>
<td>⊕</td>
<td>⊖</td>
<td>⊖</td>
</tr>
<tr>
<td>16. Köcher et al. (2020b) &amp; Köcher et al. (2020a)</td>
<td>•</td>
<td>•</td>
<td>Yes</td>
<td>⊥</td>
<td>⊕</td>
<td>⊖</td>
</tr>
</tbody>
</table>

<sup>a</sup> • = initial scope, ◦ = extended scope; <sup>b</sup> ⊕ = specific, ⊖ = independent, ⊗ = both
match the skills of resources with product specifications for matchmaking purposes. Perzylo et al. (2019) semantically described manufacturing skills in a cognitive manufacturing system for matchmaking applications and introduced an approach for the automatic orchestration of higher-level skills from a given set of basic skills using their own domain specific language.

Bennulf et al. (2021) focused on the simplification of reconfigurations when new types of products should be added. Process plans of products are defined with a sequence of skills to be utilised without specifying resources. They combine skills via skill conditions, called connection requirements. When a process plan requires a skill, e.g., grip with a gripper resource, then that skill may require further skills, e.g., move with a robot resource. These requirements create a tree of connected resources (and related skills) that are not explicitly defined in the process plan. Hashemi-Petroodi et al. (2019) alternatively developed a three-step methodology to reconfigure RMSs. Feasible assembly plans are set-up using a skill-based comparison of the product requirements and RMS. The skill comparison is based on a solely semantic matching process. Assigned resources are only checked afterwards through a quantitative parameter analysis.

Most of the above contributions start from the product and search for all required, available processes. These processes are then translated to a list of required skills that are subsequently independent of the product. The drawback of this approach is the need for an additional step, namely, the translation of product requirements to processes. Hoang, Hildebrandt, and Fay (2018) proposed a more product-oriented approach to limit modelling and communication effort during reasoning. Because of the extensive list of manufacturing methods, they question if process information is, from a modelling point of view, an efficient interface for matching product requirements against manufacturing skills and subsequently checking the production feasibility. In Hoang and Fay (2019) and Hoang et al. (2019) the product-oriented approach is worked out using AutomationML (AML) and XML. It comprises a model specifying property spaces and an interdependency model identifying relations between the system elements. The skills of resources are represented as ranges of product property values. A drawback is that deducing the important properties can be difficult and if new product properties are introduced, a whole set of skill descriptions should be updated and reconnected to the resources. This is in contrast to the manufacturing process-oriented approach, where only new processes result in new skills and existing skills don’t change.

2.1.2. Online modelling perspective

The second category of skill-based models focuses on the execution level. They mostly aim at online interventions as task-oriented programming, autonomous set-up, disturbance handling and plug-and-produce actions. In Pfrommer et al. (2014) a service-oriented framework for modelling and the operation of adaptable manufacturing systems was developed. Their framework consists of three main components, spanning from long-term planning to the runtime orchestration of available resources during assembly. For autonomous set-ups, equipment is uniformly identified via high level skills that can be matched with the product-specific manufacturing requirements embedded in the product’s bill of processes. From these high-level descriptions, executable skills are derived and made available as services (Pfrommer et al. (2015)). In Backhaus and Reinhart (2017) a similar concept is implemented using AML, but the focus is on task-oriented programming. Here, skill descriptions are used for the vendor-independent modelling of resources, processes and products. Their taxonomy is mainly built on
five top-level skills which form the core of the model, but can be subdivided into more granular skills. Danny et al. (2017) actively developed an openly accessible plug-and-produce system using AML and Open Platform Communications Unified Architecture (OPC UA). Equipment is made self-descriptive on a service bus by the use of skills. The concept of skill recipes and skill requirements is used to execute the skills to fulfil the assembly requirements. Profanter et al. (2017) presented a system architecture that uses generic and semantically augmented OPC UA skills for robots, tools, and other system components to facilitate flexible component interchange and automatic parametrisation. Rovida and Krüger (2015) developed a methodology for task-level robot programming by providing robots with a set of movement primitives and skills. The skills operate as fundamental software blocks modifying the world state and work via high level parameters related to the requirements (e.g. end location of move). Similarly, Pedersen et al. (2016) presented a conceptual model for robot skills that are similar to the abstraction level used when instructing tasks to human workers. They see robot skills as the high-level building blocks from which a complete robot program, or task, can be composed, and that allow the robot to complete the task. More recently, Schou et al. (2018) presented a skill-based programming tool that facilitates easy and fast instruction of collaborative robots by inexperienced operators and robotic novices.

2.1.3. Multi-perspective modelling

Lastly, Köcher et al. (2020b) explicitly tried to embed all applications as specified in table 1 and de facto integrate both (semi-)offline and online perspectives into their model. Their formal model of machine skills consists of three sub-models, two of the models line up with the two modelling perspectives and related applications. The third model specifies the machine structure. They try to close the gap between high-level and executable skills and enable reuse by building on existing standards and aligning them in an alignment ontology. Nevertheless, there remains a large gap between the high-level skills and the executable skills. Current research is focused on execution applications: e.g. automated code generation to facilitate model-based set-up development (Köcher, Hayward, and Fay (2022)) and constraint checking for executable skills (Köcher, da Silva, and Fay (2021)).

2.2. Challenges

As indicated in the previous section, a lot of developments have been made concerning skill-based modelling in the past few years. The only common ground between all aforementioned models is the uncoupling between products (and related sub-goals) and resources via skills. Some models focus on matchmaking between product definitions and resources, as well as the planning of the latter. Other models make resources self-descriptive in an online environment for runtime orchestration or task-oriented programming. Therefore, most of the above models are application dependent and have different interpretations of the skill concept (table 1 and 2). Furthermore, they differ in their level of granularity, implementations and modelling languages which makes it difficult to merge and reuse them. Almost no research is multi-perspective and tries to bridge the gap between offline and online needs. Consequently, it can be concluded that various opportunities remain unused and a more integrated approach is desirable.

The challenges for a universally applicable formal skill model are multifaceted. Based on the various viewpoints (section 2.1) and discussions in literature, we identify a
formal skill model should ideally comprise five different aspects: (1) Firstly, it is important to keep skill-based models generic and extendable (Köcher et al. (2020b)). Most research is focused on a single perspective because each targeted area of activity in manufacturing has its own perception of what is needed, even though they agree on the same overall intentions. Recent literature underlines the difficulty of defining one common model for interoperable manufacturing knowledge systems and specifies the required extendability of models (Köcher et al. (2020b); Palmer et al. (2018)). (2) This extendability could be in the form of a standardised and technology-independent core skill description schema from which domain-specific and sometimes overlapping schemes can branch of (Pfrommer et al. (2015)). (3) The skill concept should be multidimensional (Malakuti et al. (2018)). On one hand matchmaking applications require to semantically integrate resource capacities with product and process requirements (Agyapong-Kodua, Haraszkó, and Németh (2014)) asking for the possibility to make an abstraction of functionalities by defining realisation-neutral skills. Skills should thus be resource-, process- and product-independent (Weser et al. (2020)). On the other hand more refined resource-, process- and product-specific skills will be required for optimal reconfigurations, (micro-)scheduling operations and task-oriented programming of specific resources. (4) Skills always describe an effect on something and thus induce change in some relation or property (Perzylo et al. (2019)). Therefore a skill model should be property-based. (5) Nevertheless skills should be more than simple property carriers, they should characterise the input, output and transient behaviour implicating the modelling of states.

3. Formal skill model

The here proposed formal skill model tries to bridge the gap between the different perspectives online and offline applications encounter and starts from the challenges identified in section 2.2. Section 3.1 first describes the development process. Section 3.2 points out the model requirements and section 3.3 gives a definition of the formal skill model and explains the main concepts.

3.1. Development process

In order to ensure a systematic development of the formal skill model, we applied the widely used methodology of Sure, Staab, and Studer (2009), which was further illustrated by Järvenpää et al. (2019). This methodology consists of five phases:

(1) The feasibility study started from the internal need for a broad skill model that delivers generic descriptions of needs and capacities with skills as the connector between products, processes and resources in assembly. The current literature was analysed, resulting in the comparison of models as discussed in section 2.
(2) During the kick-off the modelling challenges were discussed (section 2.2) and the model requirements were further clarified (section 3.2). Together with researchers in multiple fields from the strategic centre for the manufacturing industry in Flanders, a series of workshops led to a first version of the formal skill model (Van De Ginste et al. (2021)).
(3) Next, refinements of this model were made specifically for assembly. Moreover, a mapping was made to ISA-95, resulting in the here-presented formal skill model (section 3.3).
The evaluation phase is in fact a long process. The model should be evaluated for different case-specific implementations to verify its universal applicability. Here a first step is take in the evaluation on a RAS (section 4).

Finally, the last phase consists of application and evolution. Feedback to the formal model remains possible. Models require maintenance to fulfill the changing requirements during its use. Continuous feedback from both users and applications helps to improve the model. Even if the model is static, the real world is changing continuously. Through this natural evolution, the constraints may vary which could affect the model and its possible extensions.

3.2. Model requirements

To bridge the gap between the online and offline modelling perspective, it is important that the model is extendable, technology-independent, multidimensional and state-based as identified in section 2.2. However one cannot model skills with only these characteristics in mind. To further structure the requirements during the kick-off of the model development, competency questions were used and set-up during a series of workshops (Grüninger and Fax (1995)). According to literature, these competency questions can later (section 4.3) be used to further evaluate whether the model meets the requirements (Noy, McGuinness et al. (2001)).

The workshops with researchers active in multiple fields from the strategic centre for the manufacturing industry in Flanders brought as a conclusion that the model should provide an answer to general questions as 'What will (or can) happen?' and 'How should it happen?' in assembly systems.

The model should provide information on product definitions in terms of skills and on the suitability of resources for the execution of a skill. This can be translated into specific competency questions.

For the definition of the products it is important to know an answer to:

(1) What are the considered products (description)?
(2) What are the requested products (delivery date, volume)?
(3) What skills are required for the requested product(s)?

For coupling resources to skills, both the current set-up and the required configuration are important, which requires answers to:

(4) What resources (material or equipment; abstract or specific) does a skill require as prerequisites?
(5) What resources does a skill consume?
(6) What resources does a skill produce?
(7) What resources form functional combinations in a skill?
(8) What transitions on the current system configuration (e.g. combinations of resources) are required to meet the prerequisites of the required skills?

With these requirements in mind, the model is further developed ultimately resulting in the proposed formal skill model.

3.3. Model representation

Figure 1 conceptually shows the formal skill model as integrated in the Resource Description Framework (RDF). The flow diagram gives a description of the relation be-
tween all objects. Table 3 further specifies all objects/classes that are part of the model. As skills uncouple products, processes and resources, we further recognise three main parts in the model. The first part (blue) describes the resources by means of states. The following part (orange) couples the operations schedule to product definitions and skills. The last part (yellow) couples the resource-independent operations to resources in the work definitions.

3.3.1. Resource modelling in states

The basis for operations schedules and product definitions are the involved resources with their connections and properties. Here, both materials and equipment are represented and handled equally as they come together as ‘equal partners’ during assembly operations.

In general, physical and logical compatibility between resources can be checked by comparing several resource properties. For example, does the inner diameter of part A match the outer diameter of part B? However, when looking closely to assembly,
Table 3. Formal skill model classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill (Work definition)</td>
<td>An identification of the (abstract) resources and workflow required to perform a specified unit of work, abstract class (ANSI/ISA-95.00.04 (2018)). The uncoupling between products, processes and resources.</td>
</tr>
<tr>
<td>Work Master</td>
<td>Template information not associated with any specific job order, defines the detailed steps needed to accomplish all or part of the operation (ANSI/ISA-95.00.04 (2018)).</td>
</tr>
<tr>
<td>Work Directive</td>
<td>Start as copies of work masters and are augmented with information for a specific job order (for execution) (ANSI/ISA-95.00.04 (2018)).</td>
</tr>
<tr>
<td>State</td>
<td>A structure consisting out of state resources, properties and/or connections.</td>
</tr>
<tr>
<td>State Resource</td>
<td>Any resource to be part of the state.</td>
</tr>
<tr>
<td>State Connection</td>
<td>Any connection, connecting resources via state interfaces within a state.</td>
</tr>
<tr>
<td>State Property</td>
<td>Any property of importance to the state from a resource, interface or connection referred to in the other state components.</td>
</tr>
<tr>
<td>State Interface</td>
<td>Any interface, part of a state resource within a state.</td>
</tr>
<tr>
<td>Interface</td>
<td>Anchor point for connections, part of a resource.</td>
</tr>
<tr>
<td>Operations Definition</td>
<td>The information needed to quantify a segment for a specific operation (ANSI/ISA-95.00.02 (2018)).</td>
</tr>
<tr>
<td>Operations Segment</td>
<td>The ordering/sequencing rules related to the operations segment execution (ANSI/ISA-95.00.02 (2018)).</td>
</tr>
<tr>
<td>Operations Request</td>
<td>A request for an element of an operation schedule (ANSI/ISA-95.00.02 (2018)).</td>
</tr>
<tr>
<td>Segment Requirement</td>
<td>The building blocks of an operations request, corresponding or referencing an operations segment (ANSI/ISA-95.00.02 (2018)).</td>
</tr>
<tr>
<td>Job Order</td>
<td>Unit of scheduled work that is dispatched for execution (ANSI/ISA-95.00.04 (2018)).</td>
</tr>
<tr>
<td>Job Order Dependency</td>
<td>The ordering/sequencing rules related to the job order execution.</td>
</tr>
</tbody>
</table>

an assembly operation can be seen as a transition where a connection between two resources is made. A disassembly operation on the other hand will consume a connection and leave the resources untouched. Interfaces could be seen as a property of a resource, but here they are considered more important than a standard property. Therefore, connections and relations between the interfaces of resources are made explicit by the structural resource states.

As can be seen in figure 1, skills are the central element describing the applicable transitions between abstract states. The level of granularity is application-dependent. The states are a combination of resources (both material and equipment), interfaces and properties characterising both the input, output and transient behaviour of the skills. The states are seen as stable.

3.3.2. Operations schedule

Generally, Enterprise Resource Planning (ERP) systems or other specialised scheduling systems are responsible for the planning of orders. Nevertheless as stated by Jaskó et al. (2020) production planning and scheduling should stand in close connection with resource management and online orchestration ensuring the optimal usability of local resources. The presented skill model thus clearly touches the field of Manufacturing Execution Systems (MES) which monitor, control and optimise operations management activities. The ANSI/ISA-95.00.02 (2018) and ANSI/ISA-95.00.04 (2018) standard is the standard for the formalisation of data exchange between the shop floor and other layers within a manufacturing company. According to Cottyn et al. (2011), ISA-95 can serve as a common model of integration. It can be used as a standard terminology to define system requirements and integrate different software systems.

The request for operations to be performed is defined in the operations schedule
Figure 2. Four states as part of an operations segment that connects two materials

model of ISA-95. Herein, operations requests contain the information required for the production scheduling to fulfil work in time (e.g. when to start, when to be finished and the priority of the request) and are further split into segment requirements. As can be seen on figure 1, they each correspond to elements of the operations definition model. The actual scheduling of specific units of work requested for execution is defined in the work schedule model as a job order. A job order directly refers to a required skill.

Of course, one cannot schedule products without knowing the product definitions. An operations definition defines a product, each operations segment performs a transition and the operations segment dependencies result in a precedence graph. In the skill model the operations segments are also coupled with states. Together they form the product definitions representing all value-adding tasks for the requested products. Figure 2 depicts an example of the four states that are part of an operations segment that connects part A to part B via a physical connection. Material A and B can both be abstract (e.g. screw) or more specific (e.g. screw M6). The level of abstractness can thus be easily chosen depending on the linked resource.

3.3.3. Work definition

Requested operations as in figure 2 should be matched to actual work via state transitions. The last analogy between this model and ISA-95 can therefore be found in the skill concept itself, more specifically in the work definition model. A skill (work definition in ISA-95) is an identification of the resources and workflow required to perform a specified unit of work. They translate operations segments to actual work. Work directives are associated to a specific job order, whereas work masters are not, as they solely contain the template information. The skill concept is multidimensional as there is a distinction between template and specific job information. Furthermore, skills are defined by states which can have different levels of abstraction and therefore be more or less resource-, process- or product-specific. Figure 3(a) depicts the states for a skill that connects part A and B via a set-up containing a table, a robot and a gripper. A more specific skill containing the type of gripper and robot (e.g. 3-finger gripper with articulated robot), could be added to be more resource-specific. Nevertheless all of these resources and connections are required for the skill to take place. Once the work is allocated to a set of resource instances a work directive can be derived from the work master. Instead of the abstract classes in the work master, the work directive will only contain states with objects from the instance level as a world state would reflect.

While skills respond to operations segments via the requested state transitions, they
also require a certain state of the resources. The value adding skill depicted in figure 3(a) requires a gripper connected to the robot. If this connection is not yet made in the real world, a reconfiguration is required. For these non-value adding tasks there is a need for non-value adding skills to close the gap between the current world state and the partial world state, required to enable the value adding skill. Let’s assume the gripper is located on the table that is supporting the robot and a non-value adding skill exists that can mount this gripper on an empty robot, see figure 3(b). This non-value adding skill must be used to create the required set-up.

In summary, the workflow for the coupling of production requirements to resources is as follows. The operations schedule defines the segment requirements (set of value adding tasks). The segment requirements refer to operations segments, where the requested transition is defined. The skills that can deliver this transition can be searched and the nearest (location and state) resource pool to deliver the transition can be deduced for the real-world set-up. Next missing state connections and properties for the value adding skill on the real-world set-up can be checked. Non-value adding work directives will be planned during execution. They are planned in the correct order and ahead of the value adding work directive to reconfigure the set-up where required.
4. Case-based evaluation

The real-world applicability of the formal skill model as described and illustrated in section 3 strongly depends on the depth and effort done to implement the model for the intended case. Are all skills and operations definitions modelled? Is the model properly implemented for the technologies on which systems rely? As the model is completely technology-independent, a choice of data format should be made to further use the model. Here an implementation in AML is proposed to evaluate the framework on a RAS. Section 4.1 presents the evaluation case. Section 4.2 discusses the instantiation of the model implementation in AML. Section 4.3 further formally validates if the model meets the requirements.

4.1. Reconfigurable assembly system

The proposed evaluation case (Figure 4) is a RAS as part of the infrastructure project ‘InfraFlex_Infra’ funded by the strategic centre for the manufacturing industry in Flanders. The goal of the project is to create a practical and industrially-relevant test facility for a wide range of new technologies and methods, targeting the assembly and disassembly of medium-sized products (maximal length, width and height of 30 cm, maximal weight of 30 kg). RASs are a type of changeable manufacturing systems requiring various (physical) enablers for its reconfigurability (ElMaraghy and Wiendahl (2009)): (1) The RAS evaluation case here counts a number of modular cells. Each cell consists of a base allowing the connection of various standardised add-ons. A robot is placed on top of each base to execute tasks on the connected add-ons. (2) These add-ons allow the convertibility of the set-up and add skills to the cell. Some examples of add-ons are: an operator assembly station, a part warehouse, a bin picking station, an automated part feeder and a tool change unit. Small reconfigurations are possible via skills of add-ons, e.g. robot tool changes happen via an automatic tool changer mounted on an add-on. (3) The set-up is scalable as cells stand on their own and bases can be dynamically extended by add-ons. Transport between cells can be done by Autonomous Mobile Robots (AMR) (or con-
veyors). (4) The skills of the set-up are mobile as complete add-ons can be interchanged. This means that large reconfigurations are mostly done by rearranging add-ons. As can be seen in figure 4, these add-ons are also switched via AMRs. (5) Lastly, to be automatable, the set-up is linked to the here presented skill model. The set-up is all-encompassing and comprises both matchmaking, production planning, runtime orchestration and task-oriented programming applications. Complete knowledge bases and rules, and accurate models of the changeability at macro- and micro-levels are required to automate the configuration in each of the applications.

4.2. Model instantiation

The data format AML is chosen for the model instantiation because it is a neutral data format, which has been designed to provide a seamless exchange and storage of engineering information from different engineering disciplines (Drath (2021)). It is based on the Computer Aided Engineering Exchange 3.0 (CAEX 3.0) standard, which defines a neutral data format for the storage of object hierarchies and object relations, encoded in XML (Wally (2021)). The key idea of AML is to interconnect different, already established, data formats and to define the modeling rules for mapping and interconnecting the various aspects of plant engineering.

Since AML is able to house both resource topologies and states as well as ISA-95 objects, this standard is a logical choice as the main data format for exchange of data. Figure 5 illustrates the exchange of model data for flow optimisations using the third-party FlexSim simulation software. A custom configuration tool is able to import or export configurations of products and equipment. This configurator translates all skills and the related pool of state transitions into an AML file. Model-based scheduling software is set-up to schedule the skills and send them to the simulation software. The model aids to structure the information flow and keeps the complexity manageable.
4.3. Response to competency questions

Next to a case-based instantiation and test in software environments, it is also important to formally check if the model satisfies the requirements and fulfils its promises. Table 4 therefore illustrates how the information needed to answer the competency questions from section 3.2 can be retrieved from the model. The table shows that all questions can be answered directly from the model.

5. Discussion

The formal skill model described in section 3 focuses on assembly systems and used the challenges identified in section 2.2 as a starting point. In this way, the basic elements that are required to facilitate decision support on hierarchical levels in assembly systems, are taken into account during the development of the model. The model is generic and extendable as it integrates with the ISA-95 standard for a more structured interface between the enterprise and control systems. Furthermore, it is technology-independent and is open to integrations in various data formats as here illustrated with AML. It is not bound to a specific technology. The downside is that an initial effort is required for the integration into technology specific environments. The model further differs and contributes to the research on skill-based modelling in underlining the importance of connections between resources (both material and equipment) in a complex assembly environment.

As seen in section 4, the evaluation case is an all-encompassing case with a broad spectrum of applications. This broad spectrum with applications as matchmaking, planning, runtime orchestration and task-oriented programming aids to evaluate and determine the main model features and its automatability. Section 4.3 already compared the provided competencies to the information the model can deliver. Table 5 further summarises the main model features facilitating these competencies. The table extracts the different model features from the evaluation case and puts them next to the applications the evaluation case encompasses.

In the formal skill model, the structural resource states are a combination of resources (both material and equipment), interfaces and properties characterising both the input, output and transient behaviour of the skills. This allows for both the mapping of operations definitions (product) to equipment independent state transitions (material and connections), as for the mapping of operations segments (product) to work masters (equipment) via state transitions. Connections between interfaces are modelled as 'first-class' citizens and skills are composed via resources (instead of semantic composition). The model clearly focuses on the uniform representation of resources and their connections as prerequisites and results of skills. The connections further allow the concurrent use of resources via the occupation strategy specified in the work master states. This focus on interconnections is ideal for assembly and set-ups confronted with frequent reconfigurations.

Moreover, there is also a distinction between template and specific job information via work masters and work directives. Via these work directives, the structural resource states further allow to formally and universally reflect the impact of skills on the world state of the set-up. As the structural resource states are the main focus of this model, complexity is not bound to the model, but to the application. Skills are also multidimensional as different levels of abstraction are allowed through states consisting of abstract or more specific resources resulting in more or less equipment-, product- or
Table 4. How to retrieve the information required by the competency questions?

<table>
<thead>
<tr>
<th>Competency Question</th>
<th>How the information is retrieved?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1: What are the considered products (description)?</td>
<td>Directly query from the operations definitions, consisting of operations segments with interdependencies further linked to states reflecting the product information. (Adopted from ANSI/ISA-95.00.02 (2018).)</td>
</tr>
<tr>
<td>Q2: What are the requested products (delivery date, volume)?</td>
<td>Directly query from the operations requests, consisting of segment requirements. Contains the information required for production scheduling. (Integration of ANSI/ISA-95.00.02 (2018).)</td>
</tr>
<tr>
<td>Q3: What skills are suitable for the requested product(s)?</td>
<td>Directly query the work masters linked to the corresponding operations segments.</td>
</tr>
<tr>
<td>Q4: What resources does a skill require as prerequisites?</td>
<td>Query the pre-state of the envisioned skill. Query the resources (and their connections) linked to this skill.</td>
</tr>
<tr>
<td>Q5: What resources does a skill consume?</td>
<td>Query the pre-state of the envisioned skill. Query the post-state of the envisioned skill. Compare the lists of resources for each state and determine which resources have disappeared from the pre- to the post-state.</td>
</tr>
<tr>
<td>Q6: What resources does a skill produce?</td>
<td>Query the pre-state of the envisioned skill. Query the post-state of the envisioned skill. Compare the lists of resources for each state and determine which resources are new from the pre- to the post-state.</td>
</tr>
<tr>
<td>Q7: What resources form functional combinations in a skill?</td>
<td>Query the states linked to a skill. Query the linked resources and related interfaces. Query the connections linked to these resources and interfaces, and determine the functional combinations.</td>
</tr>
<tr>
<td>Q8: What transitions on the current system configuration (e.g. combinations of resources) are required to meet the prerequisites of the required skills?</td>
<td>Query the pre-state of the envisioned skill. Query the resources, connections and related interfaces of the current setup. Determine the difference and search for work masters with a corresponding pre- and post-state to determine the assistive work masters capable of performing the required transitions.</td>
</tr>
</tbody>
</table>

Table 5. RAS evaluation case: formal skill model features and model applications

<table>
<thead>
<tr>
<th>Formal skill model feature</th>
<th>Model applications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Matchmaking for design and (re)configuration</td>
</tr>
<tr>
<td>1. Uniform representation and handling of both material and equipment structures, allowing generic reasoning</td>
<td>●</td>
</tr>
<tr>
<td>2. Connections between resources are modelled as 'first-class' citizens</td>
<td>●</td>
</tr>
<tr>
<td>3. Mapping of operations definitions (product) to equipment independent state transitions (material &amp; connections)</td>
<td>●</td>
</tr>
<tr>
<td>4. Mapping of operations segments (product) to work masters (equipment) via state transitions</td>
<td>●</td>
</tr>
<tr>
<td>5. Combining skills via resources and its connections in structural states (instead of the semantic composition of skills)</td>
<td>●</td>
</tr>
<tr>
<td>6. Different levels of abstraction via hierarchical work masters linked to resource classifications (abstract vs specific)</td>
<td>●</td>
</tr>
<tr>
<td>7. Convenient tracking of assembly system world state via structural resource states in work directives</td>
<td>●</td>
</tr>
<tr>
<td>8. Work master pre-states trigger completion of job order graphs with non-value adding work masters</td>
<td>●</td>
</tr>
<tr>
<td>9. Concurrent use of resource pool via occupation strategy specified in work master states</td>
<td>●</td>
</tr>
</tbody>
</table>
process-specific skills. Non-value adding skills can be automatically added to the job graph by analysing the pre-states of required work masters.

In general, we can conclude that during the tests on the evaluation case, it could be seen that the coupling of the skill-based model to the assembly system aids to structure the information flow and keeps the complexity manageable for matchmaking, planning, runtime orchestration and task-oriented programming. After an initial instantiation effort it is better known what can, should and is happening through reasoning on the model data. It should however be noted that the depth and effort done to implement the model for the intended case is crucial for the real-world applicability.

The different applications combined in the evaluation case illustrate the multi-perspective approach taken for the model development and highlight the added-value of both online and offline modelling perspectives in one model. However, for the detailed evaluation of the model and its applications, the all-encompassing evaluation case has some drawbacks. The evaluation case here described evaluates the main model features and their usability, but does not fully stress test each feature in each application.

The current evaluation case has shown that, to fully stress test the model features, each model application has different requirements for in-depth validation: (1) For matchmaking applications, a specific validation case should focus on greenfield workspace design with large product families and reconfigurable equipment and tools. The current evaluation case mainly focused on the matchmaking for reconfigurations, and less on greenfield design. (2) For production planning, a case with a high product mix, low volumes and frequent mix changes will further stress test the model. (3) For runtime orchestration, a specific case with a broad resource set, a lot of execution time variability and disturbances (breakdown, rush orders) is required. (4) For task-oriented programming, a specific case with a large set of skills, frequent product design changes and regularly new products launches is appropriate.

In different research projects, our research group is currently involved in, cases specific for each of the model applications emerge. They will be used during the coming year to further evaluate and validate the model. This more in-depth validation for each model application in specifically selected industrial use cases will require an enrichment effort for all embedded taxonomies. Nevertheless, this will further highlight the added value as the model provides a stable structure with the necessary anchors for the application data. For example, the importance of the connections limit model complexity by putting the focus on the relations between resources during assembly, and less on individual resource properties.

6. Conclusions and outlook

Nowadays, formal models play an essential role in various manufacturing domains. In assembly systems formal information modelling becomes even more of a focus to tackle mass customisation needs and keep complexity manageable.

The contribution described in this paper is a formal skill model designed to deliver generic descriptions of needs and capacities with skills as the connector between products, processes and resources. Furthermore, the model combines concepts coming from both offline and online modelling perspectives.

During the kick-off of the model development, an extensive literature review and analysis gave answers to the first research question wondering what elements form the basis of a formal skill model to facilitate decision support on hierarchical levels
in assembly. The need for an integrated multi-perspective skill model being generic and extendable, technology-independent, open to the multi-dimensional skill concept and characterising both the input, output and transient behaviour of the system’s properties is recognised.

Next, the model requirements were made explicit to determine how a formal skill model for assembly systems should look like. A multi-perspective skill model should answer questions as ‘what will (or can) happen?’ and ‘how should it happen?’ in assembly systems. Subsequently, modelling workshops with experts from various domains led to a first version of the model which was then further refined, specifically for assembly and mapped to ISA-95 models.

Finally, a case-based evaluation approach was followed to answer the last research question and determine how well the proposed model behaves in practice on a RAS. The skills in the formal model reflect applicable transitions on the world state, made explicit via the changing connections between resources. The evaluation showed the importance of these structural resource states consisting of resource structures. In the connections between materials and equipment for assembly both the online and offline modelling perspectives come together. The structure of the resources and the states are the enabler, the properties and additional parametric information add the context towards optimisations. The consequence is that specific model features match with specific applications present in assembly systems, further highlighting the multi-perspective approach taken.

The resource structures embedded in the prerequisites of a skill, enable matchmaking of resources for workspace design and reconfiguration. The mapping of the model to standardised ISA-95 models couples production needs to the resources allowing for more optimal production planning and a structured interface between enterprise and control systems. The possibility to couple states to the assembly environment allows for optimal runtime orchestration. Finally, the stable states, showing the prerequisites and added-value of a skill, make the model suitable for skill-based programming. In general, it is better known what can, should and is happening.

The evaluation case here described evaluates the main model features and their usability but future research should further evaluate the formal skill-based modelling approach for other industrial use cases and compare it more in-depth with other (partial) frameworks reported in literature. Different industrial use case should focus on specific applications for an in-depth evaluation and validation of specific model features. In ongoing research projects, different industrial use cases for each of the model applications emerge and will be used during the coming year to further evaluate and validate the model. Other implementations (e.g. in OPC UA) will also evaluate the concepts of the model and create a chain of supporting tools. Despite the fact that the model delivers its promises in a RAS, it can be concluded that additional evaluation steps are still to be taken to consolidate the model and come to mature model implementations. Also model maintenance and interoperability between use cases and supporting software tools is a point of attention.

Lastly, notwithstanding the technological independence and limited complexity overhead envisioned in the formal skill modelling framework, it cannot be denied that modelling and implementing all relevant aspects of the processes, resources and products in the skill model remains far from a trivial task. This additional effort is however a drawback of each formal model. The strength of this model is found in reuse of model data across the whole spectrum of applications hence the planned research and developments on a supporting tool chain.
Acknowledgements

This research is linked to the infrastructure project ‘Infrastructure for flexible assembly and disassembly of medium-sized products’ (InfraFlexInfra), funded by Flanders Make, the strategic centre for the manufacturing industry in Flanders, Belgium.

Disclosure of interest

The authors report there are no competing interests to declare.

Data availability statement

The data that support the findings of this study are available from the corresponding author, L. Van De Ginste, upon reasonable request.

References


of assembly systems 4.0: systematic literature review and research agenda.” *International Journal of Production Research* 0 (0): 1–27.


List of figures

Figure 1

- Caption: Conceptual flow diagram of the formal skill model (Based on Van De Ginste et al. (2021))
- Alt Text: Flow diagram of skill model with as central element ‘Skill’, refined into work masters and work directives. Work masters are linked to operations segments and operations definitions, both corresponding to elements from the ISA-95 operations schedule model. Work directives also reference a job order corresponding to an element in the ISA-95 work schedule model. Skills are made up out of states. States are structures consisting out of state resources, properties, interfaces and/or connections.

Figure 2

- Caption: Four states as part of an operations segment that connects two materials
- Alt Text: Four states as part of an operations segment that connects part A to part B via a physical connection. From the pre- to the post-state a physical connection ‘tightened’ is produced.

Figure 3

- Caption: Four states as part of a skill
- Alt Text: Figure 3(a) depicts the states for a skill that connects part A and B via a set-up containing a table, a robot and a gripper. From the pre- to the post-state: (1) a physical connection is consumed between the table and the part A and (2) a physical connection is produced between part A and part B. Figure 3(b) depicts the states for a skill that can be used for reconfigurations (non-value adding). Here specifically from the pre- to the post-state the gripper is physically connected to the robot and decoupled from the table.

Figure 4

- Caption: Reconfigurable assembly facility capable of assembling and disassembling a product mix of different medium-sized industrial products
- Alt Text: 3D representation of a two cell reconfigurable assembly facility. Each cell has a central robot. The first cell has an operator assembly and bin picking station. The second cell is an automated cell with a part feeder and storage unit. An add-on with parts is transported between the cells with an AMR.

Figure 5

- Caption: AML model integrations for flow optimisations
- Alt Text: AML model integration where a RAS and product configurator exchanges operations requests, the initial world state, product definitions and work masters to the work master scheduler. Both tools next exchange the RAS set-up and job order via AML into 3D simulation software analysing flows.