Skills in Assembly

Institute for Machine Tools and Industrial Management (*iwb*)
Technical University of Munich

Boltzmannstrasse 15
85748 Garching

Dipl.-Ing. Veit Hammerstingl
Phone: +49.(0)89.289.15533
E-Mail: Veit.Hammerstingl@tum.de

This work was carried out as part of the research project AKOMI (AK⊗MI) at the Institute for Machine Tools and Industrial Management (*iwb*), Technical University of Munich (TUM). Further information under: www.iwb.mw.tum.de

Citation

BibTeX / BibLaTeX:

```
@misc{Hammerstingl.2018,
  author = {Hammerstingl, Veit and Reinhart, Gunther},
  title = {Skills in Assembly},
  year = {2018},
  note = {Version 1.1},
  url = {https://mediatum.ub.tum.de/1428286},
  urldate = {2018-01-09}
}
```

License

About the research project

The aim of the research project AK⊗MI is to increase the reconfigurability of assembly systems. Using Plug & Produce approaches, operating resources can be ad-hoc connected and configured automatically. The concept is based on standard industrial components, which are automatically extended with a digital capability description (Part of a *Digital Twin*).

Project Duration: 09/2012 – 12/2015

Sponsor	Executing Organisation
Gefördert durch Bayerisches Staatsministerium für Wirtschaft, Infrastruktur, Verkehr und Technologie	VDI VDE IT

Contents

1 Introduction 4

2 Definitions 6

3 Basic Skills 9

- 3.1 Method for Identification of Basic Skills 9
- 3.2 Taxonomy 10
  - 3.2.1 Standardized Basic Skills in Assembly 10
  - 3.2.2 Configuration Skills 13
  - 3.2.3 User-specific Basic Skills 14
- 3.3 Explanation of the Taxonomy Structure 14
- 3.4 Related Sources 17
  - 3.4.1 Books and Standards 17
  - 3.4.2 Dissertations 17
  - 3.4.3 Publications 18

4 Parameters of Basic Skills 19

- 4.1 Fundamental Parameters 19
  - 4.1.1 Process-related Parameters 19
  - 4.1.2 Product-related Parameters 20
- 4.2 Used methodology 21
- 4.3 List of fundamental Skill Parameters 23
  - 4.3.1 Fundamental Parameters of Actuator Skills 23
  - 4.3.2 Fundamental Parameters of Sensor Skills 24
  - 4.3.3 Standardised Skills 25
  - 4.3.4 Configuration Skills 30

5 Composite Skills 31

- 5.1 Method for Identification of Composite Skills 32
- 5.2 Overview 32
- 5.3 Parameters of Composite Skills 34

6 Implementation 35

7 Bibliography 38

8 Review 41

1. Introduction

One of the greatest challenges of digitization is the semantic description of data. Only with a uniform understanding of the meaning of data can information be exchanged interoperably between systems.

In the field of automation technology, intensive work is being done on the implementation of Service Oriented Architectures (SOA). In this context, devices offer their capabilities as hardware-independent services in the network. With a semantic understanding, other systems can search for necessary functionalities and use them to fulfill their tasks. A functional consideration of both the production process and the operating resources gives rise to new flexibility in the planning and application of resources. A solution-neutral description of production equipment capabilities (so called *Skills*) enables approaches for automated resource planning as well as task-oriented configuration and programming.

With the widespread introduction of OPC UA into a large number of hardware components, a technological basis for this vision seems to have been created. Only a comprehensive semantic definition of the capabilities of automation components is still missing. Although individual information models already exist, the services defined in them usually represent a device model specific view and are not suitable for a solution-neutral description of manufacturing processes.

This motivates the current paper to provide an overview of solution-neutral capabilities (respectively "Skills") in automation technology.

In addition to a mere listing of Skills, the definition of necessary parameters is just as important. Without specification of a parameter set, most of the existing capabilities cannot be mapped to real hardware components, since there are no influencing factors definied that are required for execution. This issue was addressed only marginally in past scientific works. *Therefore the second part of this paper deals with this topic.*

The described skills are part of a virtual representation ("Digital Twin") of automation devices. Further properties of the presented Digital Twin are *Status*, *Identification*, *Location Data*, *Physical and Kinematic Description* as well as *Safety* aspects. Additional information can be found in Hammerstingl et al. (2016) as well as in Backhaus, Hammerstingl et al. 2017.

This work focuses on the field of assembly technology. Due to the variety and range of capabilities, the presented taxonomy can only offer an extract. Based on the developed taxonomy, it is estimated that about **300 Skills** exist, which consist approximately equally of actuator and sensory elements. Many of the skills possess completely different parameter classes (cf. Chapter 4), which reveals the effort required for a definition and presentation. The comprehensive definition of all capabilities and their parameters is therefore seen as the task of standardisation committees or the community.

This work has the following aims:

- Definition of frequently used terms in this domain (e.g. Skill)
- Creation of a consolidated skills taxonomy in assembly technology
- Creation of a consciousness for previously underrepresented skill classes (e.g. sensors, display elements, etc.)
- Overview of skill-specific parameters
- Methodology for the derivation of further skill parameters
- Preparation for future standardization
- Basis for discussion

2. Definitions

(Basic) Skill

Solution-neutral capability offered by a device, which is described from a task-specific point of view (process view)[1]. In this connection "solutions-neutral" means a description form independent of manufacturer and construction form. Skills are elementary, therefore cannot be further broken down into sub-skills. Skills are offered by a single controllable component (e.g. gripper, sensor, frequency converter). Skills have a specific set of input and output parameters and can be automatically matched with *Process Steps* (and their requirements). Examples of skills are *Grasp*, *Check Presence*, *Move*, *Screw*, etc.

The distiction between Basic and Composite Skills is sometimes hard to make, since through device cooperation many skills can be broken down into smaller sub-skills. Regarding assembly systems there can be found a set of basic devices, nonetheless (e.g. conveyors, grippers, light barriers). Therefore a Basic Skill Taxonomy is still applicable.

Composite Skill

Complexer solution-neutral capability offered by a combination of devices, which is described from a task-specific point of view (process view). In this matter "solutions-neutral" means a description form independent of manufacturer- and hardware-design. Composite Skills consist of a sequence of Basic Skills and can be automatically compared with *Tasks* (and their requirements). They have specific input and output parameters, which can differ from the individual parameters of Basic Skills. As an example, the control component of an assembly cell could offer the Composite Skill *Mount Lamp*. The control program defines a sequence of Basic Skills (cf. Chapter 5) which are offered and executed by the individual components in the cell (robots, transport equipment, etc.).

Process Step

An elementary operation of a process flow which cannot be further decomposed into substeps. Production processes can be described on the basis of Process Steps (and Tasks). Process Steps correlate semantically with device skills, on this basis an automated matchmaking between the two can take place. For this reason, they can also be described as

[1] For the first time, Morrow and Khosla (1997) describe *Skills* (of robots) as providing "higher-quality functionalities". Cândido and Barata (2007) refer to skills in production engineering, as actions that support the manufacturing process.

process-related skills. The characteristics of the defined parameters (requirements of the process) can differ from the offered device properties (of the skills). This results in the necessity of a dynamic suitability matching between process description and device capabilities.

Task

Process flow composed of a sequence of elementary Process Steps. Tasks correlate semantically with *Composite Skills*, on this basis an automated matchmaking can take place. Tasks can be composed to higher level Tasks. Production processes can be described on the basis of Tasks (and Process Steps). The characteristics of the defined parameters (requirements of the Task) can differ from the device capabilities offered. This results in the need for a dynamic suitability matching between Tasks and Composite Skills.

Service

A service is the *software implementation* of an Basic or Composite Skill, for example by means of a service-oriented architecture such as OPC UA.

Figure 1 shows the related metamodel.

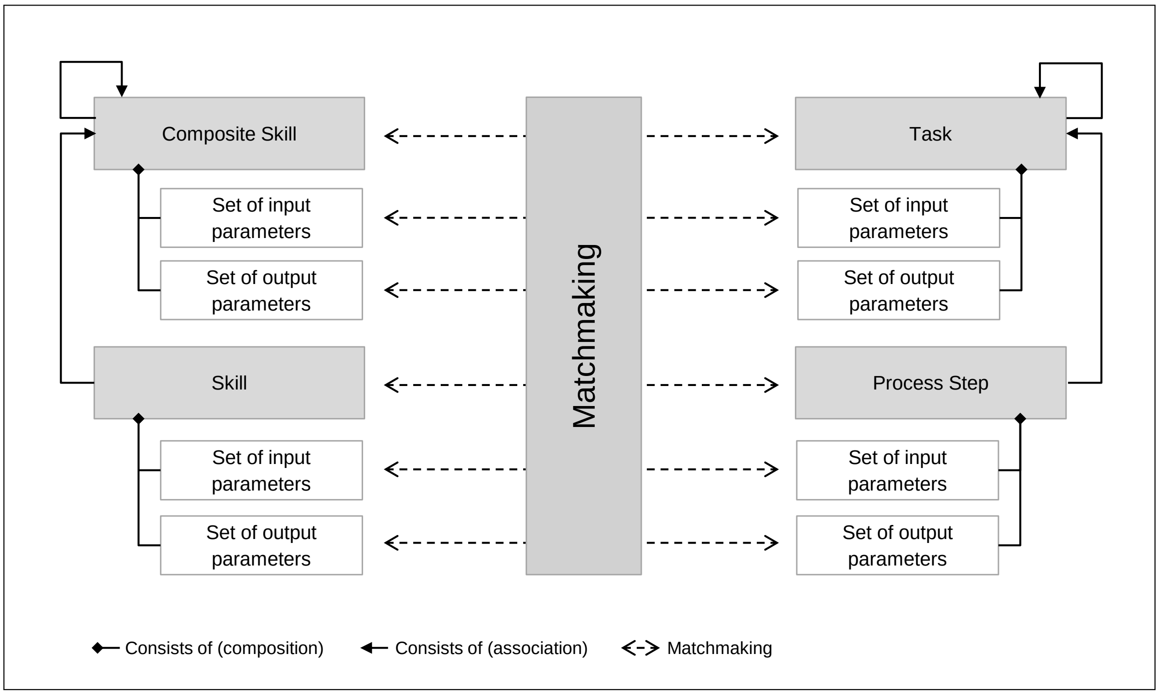

Figure 1 Skills metamodel

Note: Due to the semantic equality between skills & process steps or composite skills & tasks, there are two (instead of four) taxonomies, which can be used either on the process or on the device side, depending on the use case.

3. Basic Skills

3.1. Method for Identification of Basic Skills

The definied taxonomy is a synthesis of different sources (see section 3.4). In order to obtain this information, we searched nationally and internationally for publications, standards and dissertations in the field of capability taxonomies and ontologies.

The list of assembly functions from Lotter and Wiendahl (2012) form the basis auf this taxonomy as well as the norms VDI 2860 and DIN 8593-4. These were translated, structurally consolidated and in some areas significantly expanded (e.g. in the area of inspection operations). On the basis of the term definitions (Chapter 2) Basic Skills were separated more strongly from Composite Skills – also in their semantics.

By means of a comparison between devices in the AKOMI demonstrator and collected skills, further general skills (e.g. human-machine-interaction) as well as device-specific skills were developed.

Further explanations on individual branches are given in section 3.3. Extracts of this taxonomy are published under Hammerstingl et al. (2015) and Backhaus, Hammerstingl et al. (2017) .

3.2. Taxonomy

Skills can have different characteristics, which are described in the following sections. Figure 2 shows an overview.

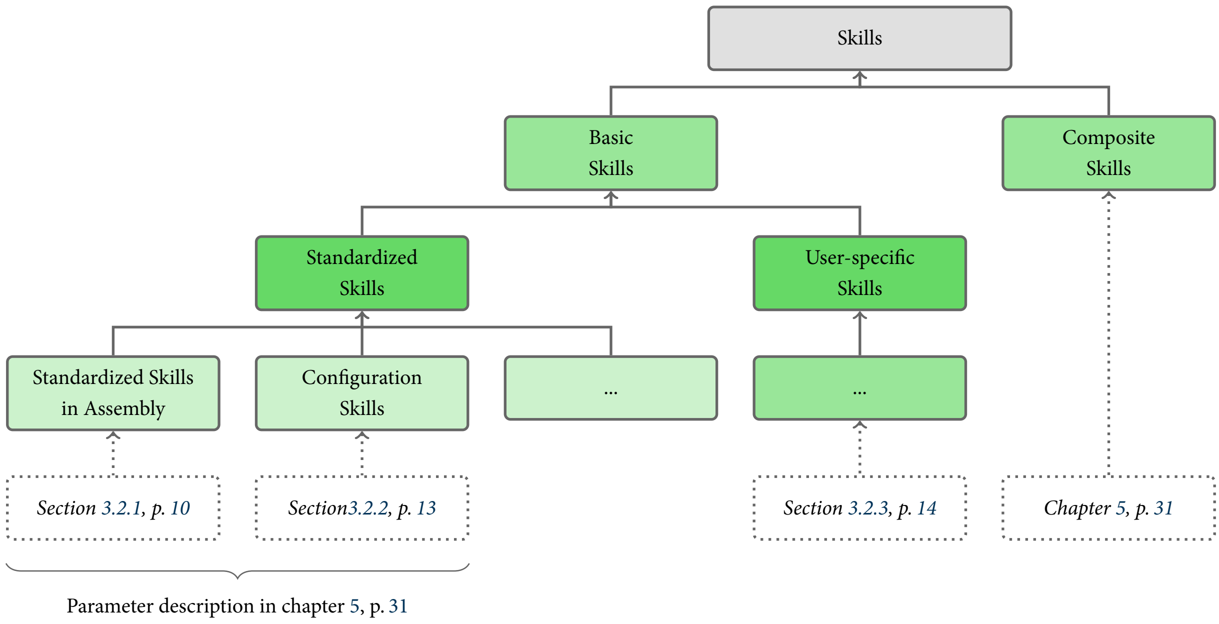

Figure 2 Characteristics of Skills

3.2.1. Standardized Basic Skills in Assembly

Figure 3 and Figure 4 provide an overview of the basic skills in assembly. The branches in the listings represent only structuring elements, the capabilities offered by devices are leaves[1].

[1] https://en.wikipedia.org/wiki/Tree_(graph_theory)

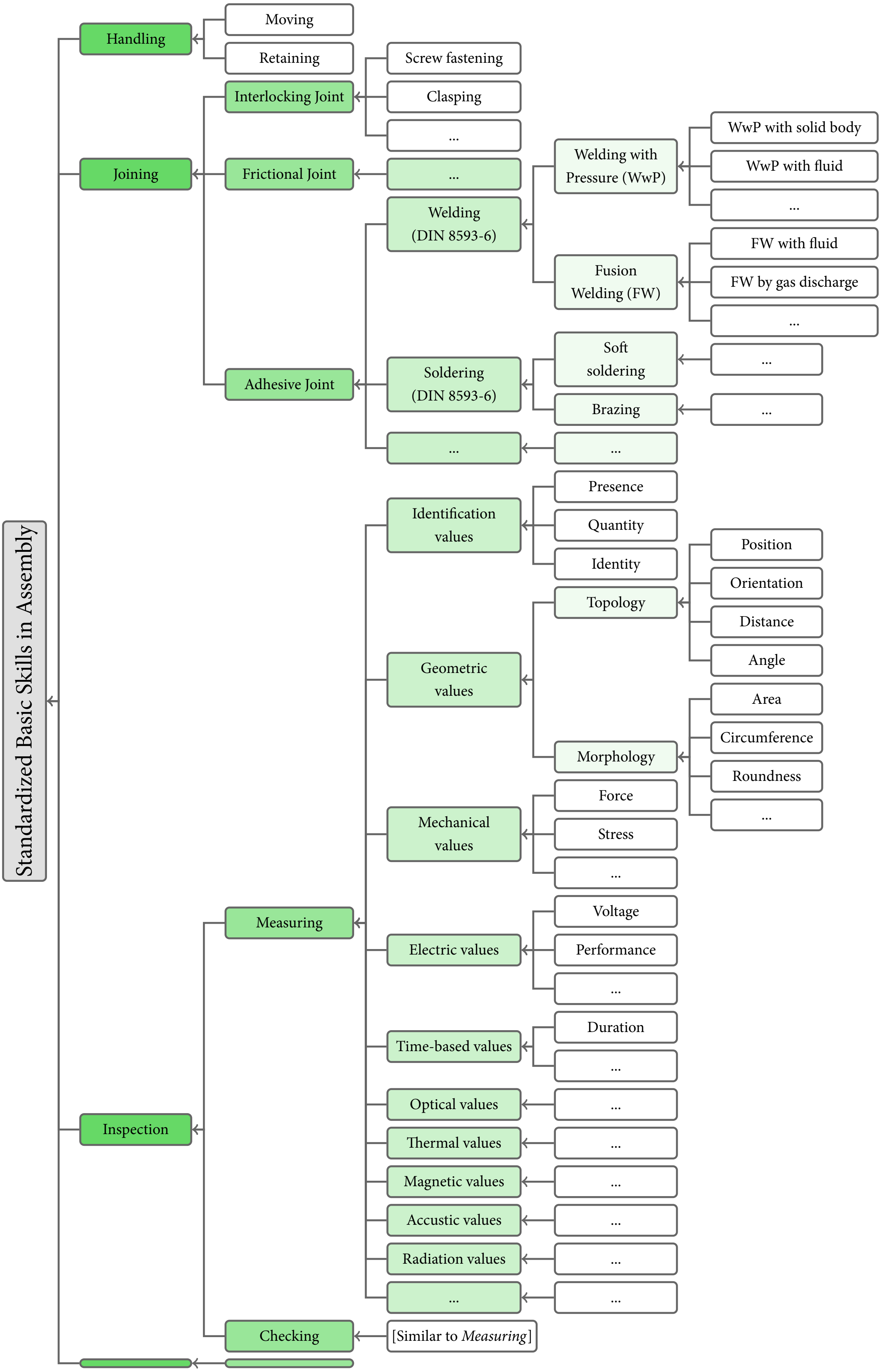

Figure 3 Standardized Basic Skills in Assembly (white)

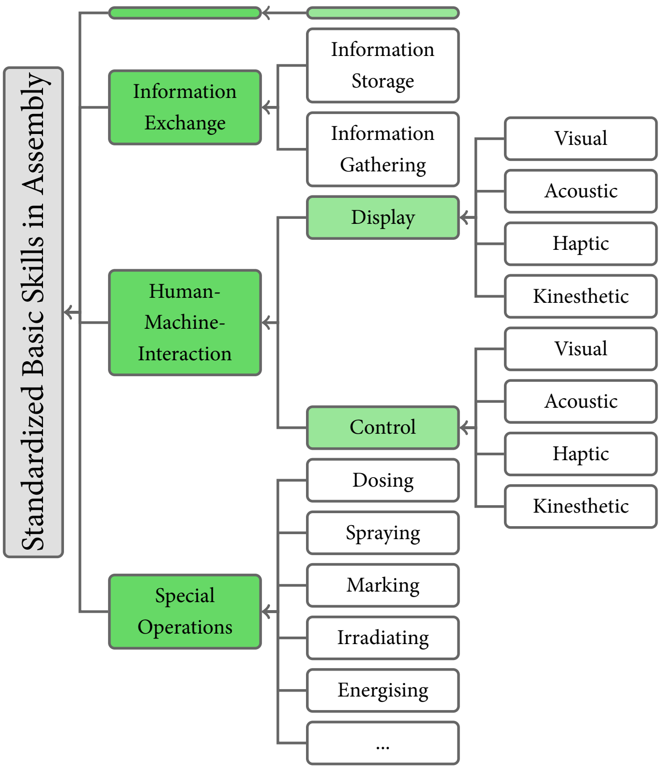

Figure 4 Standardized Basic Skills in Assembly (continued)

3.2.2. Configuration Skills

In addition to process-oriented skills, there are skills that are used for the general control of automation components. Even if these are no longer regarded solely process-based, they still meet the definition terms of manufacturer- and form-independency.

The defined configuration skills were developed from different use cases in the research project **AK⊗MI** and are presented in Figure 5.

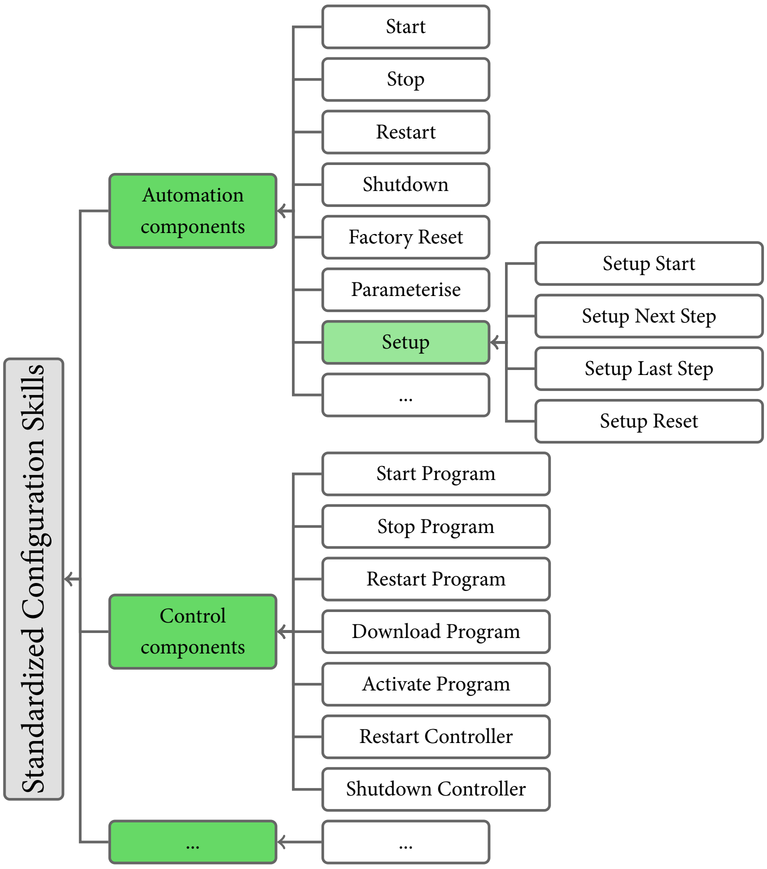

Figure 5 Standardized Configuration Skills (white)

The skill *Parameterise* enables the setup of automation devices by other software systems. The parameter characteristics are manufacturer-specific (e.g. "`set_vel_tcp=1.8`") and can be derived from technical specifications or expert knowledge. In addition, *Setup* is a skill where the device performs the setup procedure on its own with the help of a human user. For this purpose, it sends simple step-by-step work requests to the user (similar to the error handling of today's photocopiers).

Control components represent computing units (such as PLCs, IPCs or robot controllers) which take over the software coordination of the production process as well as the signal processing of primitive components.

3.2.3. User-specific Basic Skills

The taxonomy described in section **??** allows for manufacturer independent interoperability due to its semantic standardization. Of course, users can also develop their own skills which can only be implemented in a limited range of devices (for example *Measure Lid Gap Dimension*).

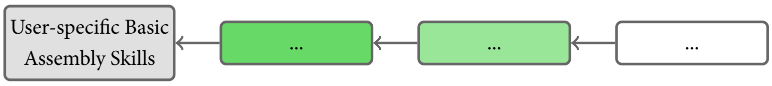

Figure 6 User-specific Basic Assembly Skills

There is the disadvantage that other software systems have only limited[2] or no understanding of the service offered by the device. On the other hand, there is the advantage of being able to define a skill specificly tailored to the user's task. At the expense of implicitly defined and implemented knowledge, the user can so potentially reduce the modelling effort. Since this approach counters the goal of achieving interoperability, it should be avoided if possible[3].

3.3. Explanation of the Taxonomy Structure

3.3.1. General Structure Information

- The overall structure retrieves extracts from Lotter and Wiendahl (2012), Backhaus and Reinhart (2015), Schmidt (1992), Hesse and Schnell (2014), Schmidtke and Jastrzebska-Fraczek (1993) as well as the standards VDI 2860 and DIN 8593-4.
- Skills can occur on different hierarchy levels. For example, while *Movement* is classified directly under the main category *Handling*, the measurement of geometrical quantities is divided into multiple layers.
- The functional process steps specified in standards are often considered to be *Tasks* and do not represent mistakes in skill taxonomy at hand. For example *Divide Quantity* is a composition of *Moving* and *Storing* and thus to be seen as a Task.

[2] Through the usage of a knowledge-based system and the classification of the newly defined skill within an ontology, partial statements can be made by automated reasoning.
[3] One example are the IEC 61131-3 programing languages, whose interoperability between the engineering tools of different manufacturers has been greatly reduced by the introduction of vendor specific modules.

3.3.2. Joining

- Particularly in the case of abilities from the *Joining*-branch it is to be noted that these can occur both as a Basic and a Composite Skill. Thus, *Fusion Welding by gas discharge* can correspond to both a single skill of a welding device and also be modeled as a Task – for example, when considering motion paths.

3.3.3. Handling

- *Moving*: Causing a material flow in a production process is also considered as an occurence of *Moving*. Often a material flow can only be created by combining several skills (e.g. *Moving* + *Securing*). In this case *Moving* occurs as a Composite Skill (respectively Task).

- *Retaining*: The capability has two dependent states, *hold* and *release*. Both states refer to a common target quantity. For modelling, it therefore makes sense that a device (e.g. gripper) offers the skill of *Retaining* and that the hold / release state is defined by its parameters. The state of retaining (holding / released) is therefore determined by its skill parameters and is not modelled as independent skills[4].

3.3.4. Inspection

- Especially the sensoric capabilites are not discussed sufficiently in available literature and therefore offer only insufficient structuring. This has been fixed in the given taxonomy.

- In order to obtain a comprehensive overview, a categorization regarding physical aspects was chosen. Due to the large number of measurable quantities (several hundred, cf. Hesse and Schnell 2014), only an excerpt can be presented here. The classification of new skills by the user is considered feasible with the given structure.

- It is assumed that each existing physical quantity shall be measured (numeric result) as well as checked (boolean result). This yields to two identical skill branches for *Measuring* and *Checking*[5].

- For the categorization of geometric quantities, the capabilities of various image processing tools were analyzed and structured in accordance with the VDI 2860 standard (cf. Hammerstingl et al. 2015).

3.3.5. Information Exchange

- Often, technical systems must use information which was collected by other systems. To do this, it is necessary to define data storage skills. For example, a database connected to a network can offer the skills *Store Information* and *Gather Information*, which can then be used by other devices.

---

[4] This is different from existing literature.
[5] But with different skill parameters, of course. See chapter 4.

- Since the information structure and semantics strongly depend on the specific application, only two generic abilities for reading and writing information can be specified here. Any semantic specialization takes place at parameter level (cf. chapter 4).

3.3.6. Human-Machine-Interaction

- Even in automated processes, users want to be able to check status information and perform correction instructions. For this reason, technical systems contain a large number of display and operation elements. In the sense of semantic interoperability, these must also be described by means of skills.

- The categorization is based on physical principles. Using the respective skill parameters, the user can further specify the display and operating mode, e. g. `Display optical information -> 2-D-bar chart`. The realisation of the display/operation is thus carried out purely on the device side.

3.3.7. Special Operations

- Many of the special functions described in Lotter and Wiendahl (2012) are more likely to be assigned to Composite Skills or respectively Tasks (e.g. *Unpack*) and are therefore missing in this list or have been semantically specified.

3.4. Related Sources

3.4.1. Books and Standards

- Lotter and Wiendahl (2012) provide a simple Basic Taxonomy based on the standards VDI 2860, DIN 8593-4 and DIN 8580. The five main groups are Joining, Handling, Inspection, Adjusting and Special Operations. The taxonomy provides a good starting point, but has the same disadvantages as the individual standards (cf. VDI 2860) due to its composition. In particular, the main category "Adjustment" is seen as a combination of skills such as Measuring, Checking and Moving, and is therefore included in the Task category. The authors make no statements on the parameters of the individual skills.

- VDI 2860: With its taxonomy, this standard also provides a good starting point. However, many of the functions shown are overlapping. For example, *Positioning* can be mapped as *Moving* and *Position Checking*. There is no structured derivation within the subcategories. Especially for the inspection category, only a flat list of few functions is provided, the corresponding measurement functions are mostly missing. No statements are made on the parameters of the individual skills.

- DIN 8593-4: This collection of standards describes a specific taxonomy for the process *joining*. The taxonomy is considered to be free of overlapping and can be transferred. Also, no statements are made about the parameter characteristics.

- DIN 8580: This standard presents comprehensive taxonomies for manufacturing processes. These do not concern the area of assembly and will not be examined further.

- Feldmann and Spur (2014): The listings in the book refer to the known sources of Lotter and Wiendahl (2012), VDI 2860 and DIN 8593-4.

3.4.2. Dissertations

- Schmidt (1992): In his work, the author refers to VDI 2860 and criticizes the overlaps of individual processes. Out of this motivation, he derives five basic assembly processes that do not overlap: Storing, Moving, Joining, Changing and Comparing. The separation of these basic processes is correct and is sometimes used as a nomenclature in the taxonomy at hand, nonetheless the vast amount of available skills requires a further structuring.

- Kluge (2011): The author describes a list of skills which is based on the well-known standards VDI 2860 and DIN 8593-4 and thus has their disadvantages.

- Loskyll (2013): The author bases the development of his functional ontology "on the directive VDI 2860 in particular" and thus has its disadvantages.

3.4.3. Publications

- Smale and Ratchev (2009): The authors describe six skill classes in assembly, without further subdivision. These are Motion, Join, Retain, Measure, Feed and Work. They correlate with the skills presented in the taxonomy at hand, despite the different categorization and nomenclature.

- Huckaby and Christensen (2012): The authors define a classification system for robot-based assembly cells. In the two-step taxonomy, skills of different kinds are classified in the same branch (e.g. Detect, Transport, Drill). The derivation of the skills is not described further. The presented skills are covered by the above mentioned standards.

- Backhaus and Reinhart (2015): The author develops a skill taxonomy based on the basic functions defined by Schmidt (1992). Underneath, he integrates the capabilities of VDI 2860 and DIN 8593-4. From all sources this is the most comprehensive representation, but here too inspection skills and special operations as well as capabilities of human-machine-interaction are excluded. The division of the skills into separating characteristics is logically structured by the author, but is interpreted differently in this work. For example, the author specifies seven different types of movement (guided linear, guided circular, unguided, rotating on the spot, etc.). In the work at hand, these characteristics are parameters of one single movement skill. This reduces the complexity of the taxonomy and by aggregating different movement parameters in one skill, combinations of movement types are possible. Backhaus and Reinhart (2015) doesn't distinguish between Basic and Composite Skills.

4. Parameters of Basic Skills

4.1. Fundamental Parameters

4.1.1. Process-related Parameters

Process Values

If a user models a process on the basis of skills, it must be possible to make specific demands on the skill, e.g. to set the *gripping force* in the skill *Retaining* or the *measurement accuracy* in the skill *Measure Distance*. These parameters are called "process-related input parameters".

Following the IPO model[1], skills also have specific output variables. These cannot be influenced by the user, but are indispensable as input variables for other components (e.g. the result of a measurement for subsequent devices).

Each process-related parameter has the following set of characteristics:

Identifier	Data type	Value *or* Value range	Unit	Necessity (option./mand.)	Input/Output

Here, the **only distinction is made between process-related skills and device skills**: If the Process Step parameters (requirements) are specified, a *value range* is defined (e. g. execution time <= 100 ms; measuring deviation < 0,1 mm)[2]. This requirement is compared with the offered *single parameter value* on the device skill side (e.g. camera system XY: measuring deviation = 0,05 mm). The specification of a single value on the process side does not appear to be reasonable, since the quantity can never be exactly met by a technical system in the mathematical sense, but is only approximated as precisely as possible.

Input parameters are generally considered to be independently definable of each other. If dependencies occur, the device or a software algorithm must carry out an optimization within the specified value ranges to find an optimal set.

[1] IPO: Input – Processing – Output
[2] This corresponds to a value range of 0 – 100 ms, respectively 0 – 0,1 mm

Geometric description

Geometry-related parameters also exist, which define *where* the skill is to be executed on the product / process. One example is the definition of gripping points / surfaces in the skill *Retaining*. These geometry-related parameters are referred to as *Region-Of-Interest (ROI)*. On the product side the ROI represents points, areas or volumes and can be modelled analytically or discretly (e.g. by means of voxel-based approaches). Although the ROI needs to be determined for the majority of skills, there are skills for which a geometric specification is irrelevant. An example of this is the *Information Exchange*.

4.1.2. Product-related Parameters

In the context of a task-oriented matchmaking, devices may be technologically unsuitable for the given process. For example, an inductive sensor can only perform a *Check Presence* skill if electrically conductive materials is present. This parameter class may not be completely covered by a purely process-related specification. This is especially the case with sensory abilities. Accordingly, there are additional Product-related Parameters (e.g. type of material) which need to be used for a technology-based matchmaking. Product-related Parameters are taken from component databases and are considered as input variables for Process Steps. Examples are mass, geometrical dimensions and material properties.

Figure 7 shows the presented parameter classes in the overview.

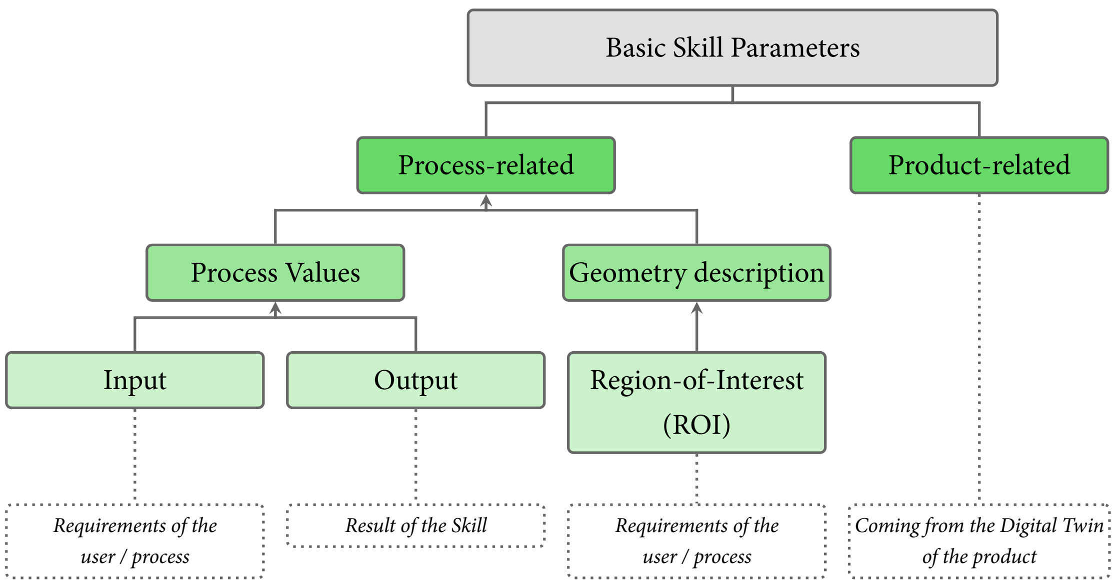

Figure 7 Overview of Basic Skills Parameters

Once the process modeling has been completed, the requirements (defined Process Steps / Tasks) can be automatically compared with the offered device capabilites (Basic Skills / Composite Skills). For this the specified skill parameters are compared with the calculated characteristics of the operating resources. Despite the syntactical and semantic equality, the so-called *matchmaking* between devices and process requirements is complex, since heterogeneous boundary conditions from different domains influence the suitability of a device. Also influencing interdependencies between cooperating devices exist.

4.2. Used methodology

Since skills represent a connection between the operating resource side and the product/process side, both domains must be considered for a parameter derivation. In order to be able to execute a specific skill correctly, devices of different manufacturers and types may require other parameters. There are two options for collecting skill-specific parameters:

Variant A: Typical devices for a skill execution are collected, such as different barcode scanners for *Identity Measuring*. The batch of all differing device parameters results in the parameter-set for the respective skill. In any case, this ensures executability on the devices, but has disadvantages: Firstly, this can result in a very large number of heterogeneous parameters. Later skill modeling requires all parameters to be specified by the user, although most of them may not be processed by the assigned device. This generates a high, partially useless modelling effort. Secondly, the semantics of the determined parameters are sometimes device-specific and not very process related. Thirdly, devices outside of the investigated set may not be able to perform the skill as necessary parameters were not considered in the definition. This may also lead to more frequent redefinitions of standardised skill parameters, which is detrimental to interoperability.

Variant B: The skill parameters are described purely process-related and physically. If a device requires further information to execute the skill, it must acquire the information by itself. This can be done, for example, by means of a teach-in concept, external software algorithms or through interaction with the user. In addition to this obligatory set, the user is also allowed to define additional optional parameters, which can be fullfilled by resources und therefore restrict the amount of suitable equipment further. This variant has the advantage that the skill parameters are in any case hardware independent, but devices must sometimes provide teach-in functionalities or a conversion logic[3]. Further advantages result from the disadvantages of variant A.

In this paper Variant B is used to define the skill parameters. For example, the parameters for inspection capabilities were derived from the DIN 1319-1. In a next stept, the determined

[3] The example of skill-based programming of machine vision systems with teach-in functionality can be taken from Hammerstingl et al. (2015) (German Paper).

parameters are validated with the necessary information of real example devices suitable for execution (cf. Figure 8, left).

In order to determine the *product-related* skill parameters, first of all possible physical principles for skill execution are listed (cf. Figure 8, right). To reduce modelling efforts, devices are then searched that implement this capability in reality. The found set is then categorized into functional principles that are suitable for the skill execution. Functional principles represent a subgroup within the respective operating principle. For each principle, the necessary physical information of the component is specified, which in sum form the product-related parameters of the respective skill.

The values for these parameters must be stored in product databases. Advantageous here is that a determined parameter value (e.g. material) can often be used for different skill parameters.

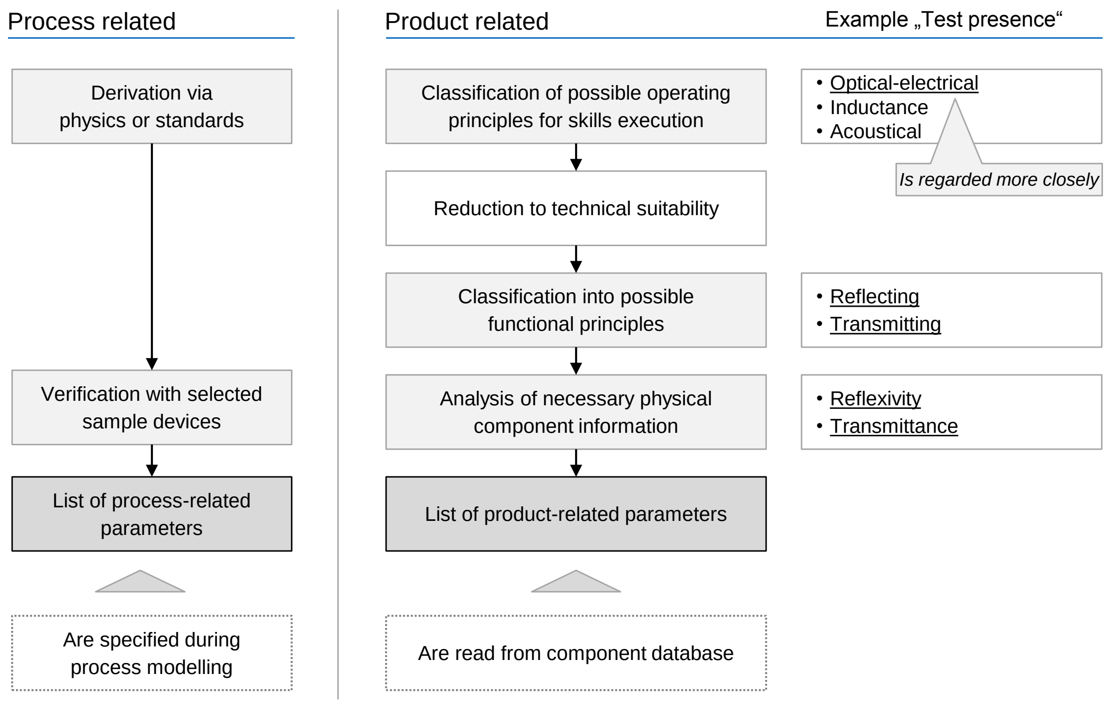

Figure 8 Methodology for determining the parameters of Basic Skills

4.3. List of fundamental Skill Parameters

The following list describes the parameters of exemplary skills. They are described from the view point of process skills, i. e. they represent requirements on the operating resources. Based on the given fundamental parameters (Section 4.3.1 and 4.3.2) and the general methodology (Section 4.2), users can define missing parameter sets themselves.

4.3.1. Fundamental Parameters of Actuator Skills

The input parameters of actuator skills are generally specific, except for the *maximum execution time* and the *region-of-interest (ROI)*. All actuator skills have the mandatory output parameters *Status* and *Progress* in common.

		Necessity	*Data type*	*Unit*
	...	...	...	...
Input	ROI	m	region	[Coordinates]
	Max. Execution Time	m	real	ms
Output	Status	m	enum	[State machine]
	Progress	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

The *Status* is implemented as a state machine, which is based on ANSI/ISA-88.00.01-2010 (cf. Figure 9). As an additional state, *not configured* was added, in which the device is initially located. By calling the setup process via the *Setup* skill, it switches to *Idle*. A factory reset can be used to return to the *not configured*-state. The state machine applies to all abilities.

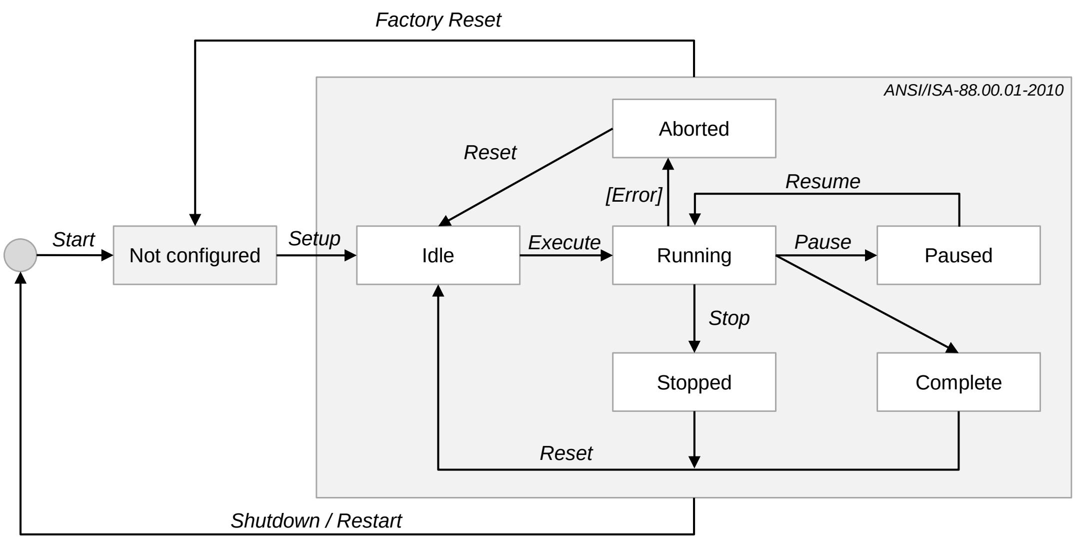

Figure 9 State machine of skills. Based on ANSI/ISA-88.00.01-2010.

4.3.2. Fundamental Parameters of Sensor Skills

During a measuring process, the *unit* of the result, the maximum *measurement uncertainty* and the maximum *execution time* must be specified. The result represents a numerical value in the given SI unit.

Skill class: Measuring		*Necessity*	*Data type*	*Unit*
Input	Unit	m	string	[SI unit]
	Measurement uncertainty	m	real	[SI unit] or percentage
	ROI	m	region	[coordinates]
	Max. Execution Time	m	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Result	m	real	[SI unit]

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

In the case of checking skills, a *check value* with SI unit must be defined. The *check condition* describes whether the current measured value must be within, outside or equal to the test quantity. Furthermore, the *check tolerance* within a test is successful (e.g. Skill *Check Distance* < 2 mm))[4] must be specified. The check result represents a boolean value.

Skill class: Checking		*Necessity*	*Data type*	*Unit*
Input	Check value	m	real	[SI unit]
	Check condition	m	string	[Operator]
	Check tolerance	m	string/real	[SI unit] or percentage
	ROI	m	region	[coordinates]
	Max. Execution Time	m	real	ms
	...	o	...	
Output	Status	m	enum	[State machine]
	Result	m	bool	true/false

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

[4] The lower bound would be 0 mm in this example.

4.3.3. Standardised Skills

Moving

Moving		*Necessity*	*Data type*	*Unit*
Input	Start pose	m	frame	[coordinates]
	End pose	m	frame	[coordinates]
	Type of movement	o	enum	[lin], [circ]
	Pose accuracy	o	vector(6)	mm or rad
	Velocity	o	real or function	m/s
	Acceleration	o	real or function	m/s^2
	Trajectory	o	frame array	[coordinates]
	Trajectory accuracy	o	vector(6) array	[coordinates]
	Max. Execution Time	m	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Progress	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Notes:

Moving has no ROI and is not modeled directly by the user. This is due to the fact that the start and end positions depend on the spatial relationship between the executing device and the product. Accordingly, the start- and end-position is device-dependent or refers to factory world coordinates. Both contradict the process-related solution neutrality. In the case of a joining operation, the ROI of the *Joining* skill would rather describe the relative positions of the joining partners; the final position of the moving skill can be calculated from the ROI of the *Retaining* skill and the pose of the *Joining* base part.

Moving has an important meaning in material flow, but in the case of matchmaking between process steps and resources, it is not described by the user but generated dynamically depending on the selected equipment.

The *Maximum Execution Time* allows the system to determine its own speed and acceleration. If these values are specified by the user, they can be defined as maximum values or be represented as a function over time. The *Trajectory* can be used to specify a path to the target pose – otherwise it is calculated by the device.

If the skill is defined overdetermined – for example by the specifications of speed as well as acceleration, accuracy and execution time – it is the software system's responsibility to carry out an optimization within the given value ranges in order to arrive at a corresponding value on the device side.

Retain

Retain		*Necessity*	*Data type*	*Unit*
Input	Active	m	bool	true/false
	Type of securing	m	enum	[standard]
	Retain force	o	real	N
	Pose Accuracy	o	vector(6)	mm or rad
	ROI	m	region	[coordinates]
	Max. Execution Time	o	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Progress	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Notes: The *Active* parameter can be used to start or stop a retaining operation. A force-locking or interlocking operating principle is selected via the type of fuse. If no force is specified, the system will retain with the maximum force.

Joining

Arc Welding (Gas discharge)		*Necessity*	*Data type*	*Unit*
Input	Active	m	bool	true/false
	Welding voltage	m	real	V
	Welding current	m	real	A
	Gas type	m	enum	[Norm]
	Gas flow rate	m	real	m^3/s
	Electrode shape	m	enum	[Standard]
	Electrode material	m	enum	[Standard]
	Wire feed speed	m	real	m/s
	Wire material	m	enum	[Standard]
	ROI	m	region	[Coordinates]
	Max. Execution Time	o	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Progress	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Notes: The raw material is derived from the product model.

Measuring

Measure Force		*Necessity*	*Data type*	*Unit*
Input	Unit	m	string	N
	Measuring uncertainty	m	real	[SI unit] or percentage
	ROI	m	region	[coordinates]
	Max. Execution Time	m	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Result	m	real	[SI unit]

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Notes: All other measurement skills are modeled similarly.

Checking

Check Presence		*Necessity*	*Data type*	*unit*
Input	Check value	m	real	[Unitless]
	Check condition	m	string	[Operator]
	Check tolerance	m	string/real	[Unitless]
	ROI	m	region	[Coordinates]
	Max. Execution Time	m	real	ms
	...	o	...	
Output	Status	m	enum	[State machine]
	Result	m	bool	true/false

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Notes: Other checking skills are modeled similarly.

Information exchange

Information Storage		*Necessity*	*Data type*	*Unit*
Input	Table	o	string	[...]
	Identification	m	string	[...]
	Key	m	string	[...]
	Value	m	string	[...]
	Max. Execution Time	m	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Progress	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Information Gathering		*Necessity*	*Data type*	*Unit*
Input	Table	o	string	[...]
	Identification	m	string	[...]
	Key	m	string	[...]
	Max. Execution Time	m	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Progress	m	real	%
	Value	m	string	–

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Notes:
For the processing of information, the semantics of the information are strongly dependent on the task or hardware. A generic skill interface can therefore only be set up if information processing is generally valid. This shifts the semantic standardization one level "lower" within the parameter values that have to be stored.

Identification uniquely identifies an entity (for example, a specific product) to which information can be attached using the key–value–relationship. The *identification* can be seen as an index which is filled with key–value–pairs. Information skills have no ROI because they store output parameters of device skills and are therefore not directly product-related.

Human-Machine-Interaction

Display Information Visually		*Necessity*	*Data type*	*Unit*
Input	Displayed value	m	[...]	[...]
	Observation time	m	enum	[1]
	Data volume	m	enum	[2]
	Extremes	o	[...]	[3]
	Max. Execution Time	m	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Progress	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Notes:
- Using display skills, it is assumed that the user only defines the generic parameters of the display and that the device generates the required display form and layout itself.
- Human-Machine-Interaction skills have no ROI because they process output parameters of device skills and are therefore not directly product-related.

- Using the data type of the *display value* (e.g. enumeration or numeric) and the other input parameters, the display device can determine a suitable display format itself. In the case of *Visual Display* skills these include analog displays, digital displays or pictorial displays with their respective characteristics as actual value display, difference display, pre-display, imperative display and more[5]. To build the displayed content accordingly, the devices sometimes require queries from the user (cf. section 3.2.2).
- [1]: Real time or historical data.
- [2]: Number of information to be displayed at the same time (1 to ∞). For example, the display of two drive speeds in one diagram results in the data range of *two*.
- [3]: Min.-max.-values for defining the display limits.

Special Operations

Heating		*Necessity*	*Data type*	*Unit*
Input	Heat flow	m	real	W
	ROI	m	region	[coordinates]
	...	o	...	...
Output	Status	m	enum	[State machine]
	Progress	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Notes:
The achievement of a required target temperature can be determined via a Composite Skill containing *Measure Temperature* and *Heating*.

[5] cf. Schmidtke and Jastrzebska-Fraczek (1993)

4.3.4. Configuration Skills

Automation Component

General		*Necessity*	*Data type*	*Unit*
Input	Delay time	o	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Progress	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Parametrization		*Necessity*	*Data type*	*Unit*
Input	Key	m	string	[...]
	Value	m	string	[...]
	Max. Execution Time	m	real	ms
	...	o	...	...
Output	Status	m	enum	[State machine]
	Progress	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Note:

In addition to skills of *Information Exchange* the skill *Parameterisation* is strongly dependent on the task and hardware. A definition must therefore be generic on the skills side. The use of a knowledge base (ontology) is recommended, which determines the specific parameters required for the device on the basis of a given task and informs the device about the parameterisation skills mentioned here.

5. Composite Skills

Composite Skills consist of an arrangement of Basic Skills. Examples are *Joining by Pressing* or *Packaging Product*. The Composite Skills parameter number and type don't need to be the same as the Basic Skills parameters. Furthermore, completely new skills can emerge through a certain combination of skills as well as the combination of their underlying resources. Following the field of psychology, this phenomenon is called *emergence*.

For example, the Basic Skill *Check presence* can be used twice to perform a (discrete) height check, for example two light barrieres can detect the heights of products moving by on a conveyor belt. However, performing the height check assumes that a) different devices are selected for the two modeled basic capabilities b) these are aligned parallel in space and c) the measurement for both capabilities is called up at the same time (cf. Figure 10, left).

Another example is the connection of two devices via their mechanical interface, for example a robot with a gripper. The combination creates a new device, which can offer a Composite Skill (here: *Handling*). Note that the retain operations at the beginning and end of the handling task are both assigned to the same (still unknown) device. This requires some kind of template concept within the Process Step modelling and Task modelling (cf. Figure 10, right).

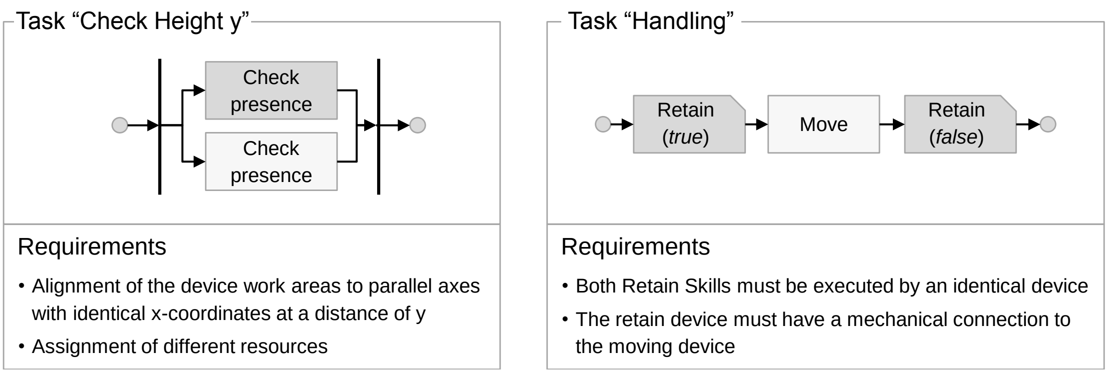

Figure 10 Examples of requirements in Task modelling

From the given examples arise the following requirements for modelling Tasks:

- Possibility of modelling control flows including parallelism based on Process Steps (Basic Skills). For example, the modelling in accordance with the sequential function chart (SFC) of IEC 61131-3 is suitable.

- Possibility to group Process Steps into Tasks.
- Possibility to define identical actors during the modelling of control flows ("Device-Templates").
- Examination of suitable devices based on their mechanical, electrical and IT interfaces as well as geometric relationships through the use of a matchmaking software.

5.1. Method for Identification of Composite Skills

The taxonomy of Composite Skills was created when identifying the taxonomy of Basic Skills (cf. chapter 3). The classification is based on the definitions described in chapter 2.

5.2. Overview

Figure 11 shows the taxonomy of Composite Skills (respectively Tasks). Many of the Composite Skills are similar in wording to the Basic Skills, but have been extended to include motion operations. Thus, the Composite Skill "Screwing" includes both the elementary process plus the associated trajectory.

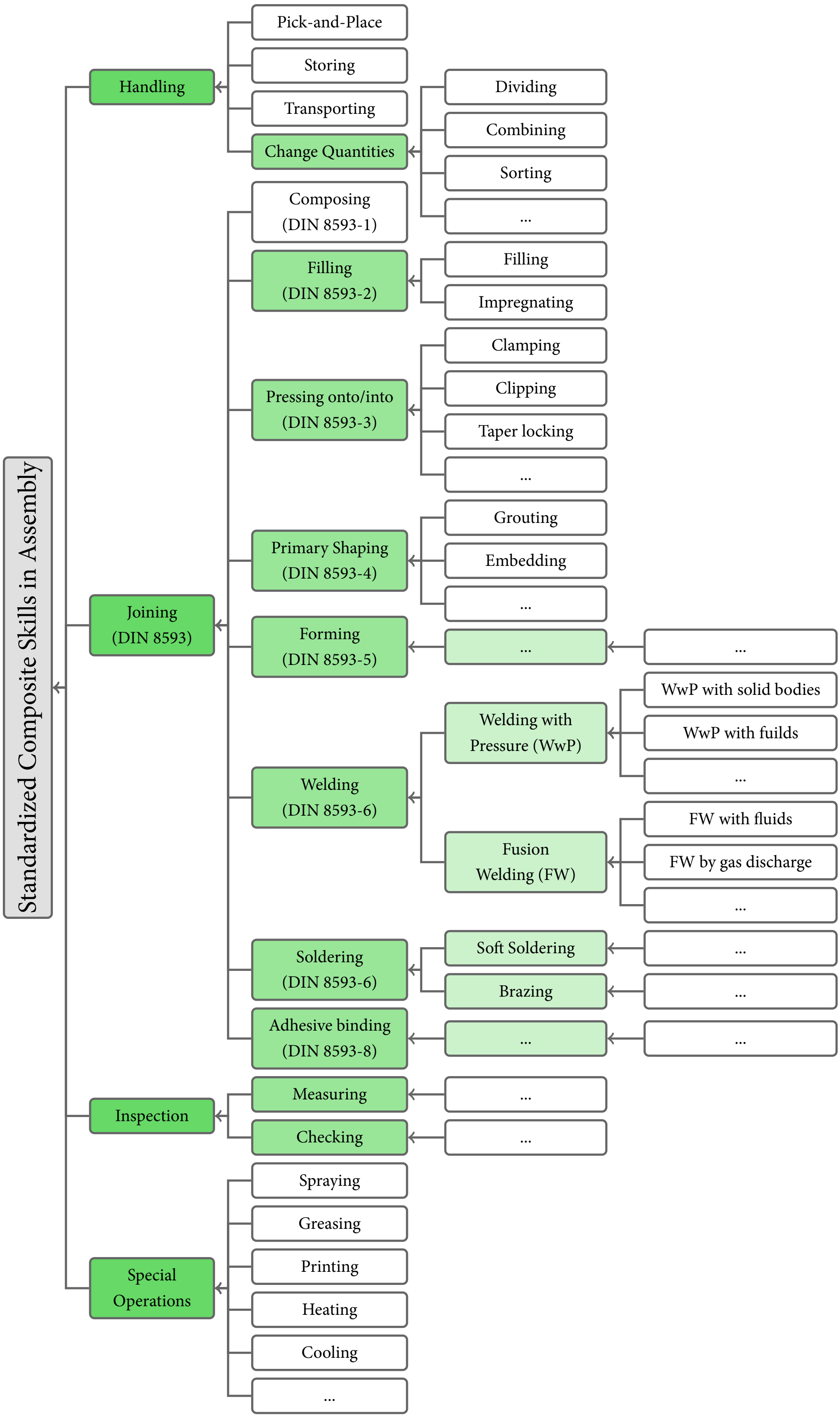

Figure 11 Standardized Composite Skills in Assembly

5.3. Parameters of Composite Skills

Storing

Storing		*Necessity*	*Data type*	*Unit*
	Ordinal Number (ON)	m	enum	[Standard]
	Storage pose	o	frame	[coordinates]
	Store	m	bool	true/false
Input	Size	o	int	[coordinates]
	Max. Lead Time	o	real	ms
	Max. Execution Time	o	real	ms
	...	o	...	...
	Status	v	enum	[Statemachine]
Output	Progress	m	real	%
	Storage Filling Level	m	real	%

All numeric input parameters are specified as ranged values. m: mandatory, o: optional

Notes:

- Storing always consists of retain and move operations and is therefore seen as a task.
- The skill has two dependent states, *Store true* (storing) and *Store false* (retrieval). Both states refer to a common set. For modelling, it therefore makes sense that a device (e.g. a buffer or warehouse) offers the skill *Storing*, and that the parameters of this device are used to define storage or retrieval[1].

[1] Thus the modelling differs from existing sources.

6. Implementation

An excerpt of the presented Basic and Composite Skills have been implemented in a software that allows the free modelling of assembly processes (cf. Figure 12). Process Steps and Tasks can be instantiated and described as sequences (logical flow). Data and material flows can be modelled independently. In each Process Step, its specific parameters can be specified.

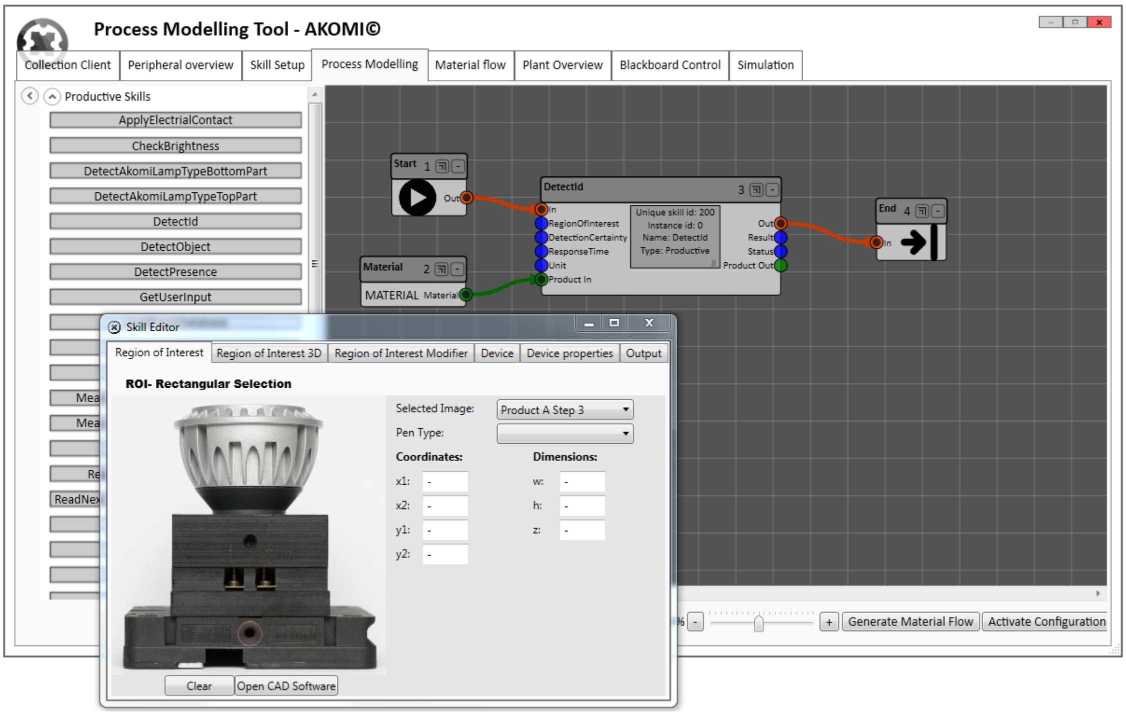

Figure 12 Screenshot of the developed software for skill-based process modelling. Left: Skills. Right: Created process sequence. Below: Detail window for the parameter specification of a Process Step.

By means of a "Blackboard Architecture", various suitability checks (e.g. process and technological suitability, geometric suitability, accessibility in the material flow, collisions etc.) are carried out for each Process Step and a matchmaking to capable production resources is created. Since capable equipment is only determined towards the end of modelling, the exact material flow is only known at that point in time. Therefore, after confirming the suggested matchmaking of primary Process Steps, it is automatically generated and inserted afterwards.

The solution-neutral process description is not only manufacturer-independent, but to a certain degree also independent of the operating principle[1]. Workers can also use smart devices to offer their Skills inside a network and be scheduled for Process Steps by the matchmaking software.

Thus, the approach enables users to increase flexibility in the area of process, volume and failure flexibility. Furthermore, it is possible to check in advance whether capable resources are available in the case of an adaption of the product or production process.

Afters the modelling, the process is then translated into structured text (IEC 61131-3) via a code generator and is automatically loaded onto a PLC. This executes the process in real time. The skills of complex devices (such as robots and vision systems) are created as services in seperate OPC UA servers. The controller then calls these services as a client.

The concept was tested on an assembly demonstrator (cf. Figure 13 and Figure 14).

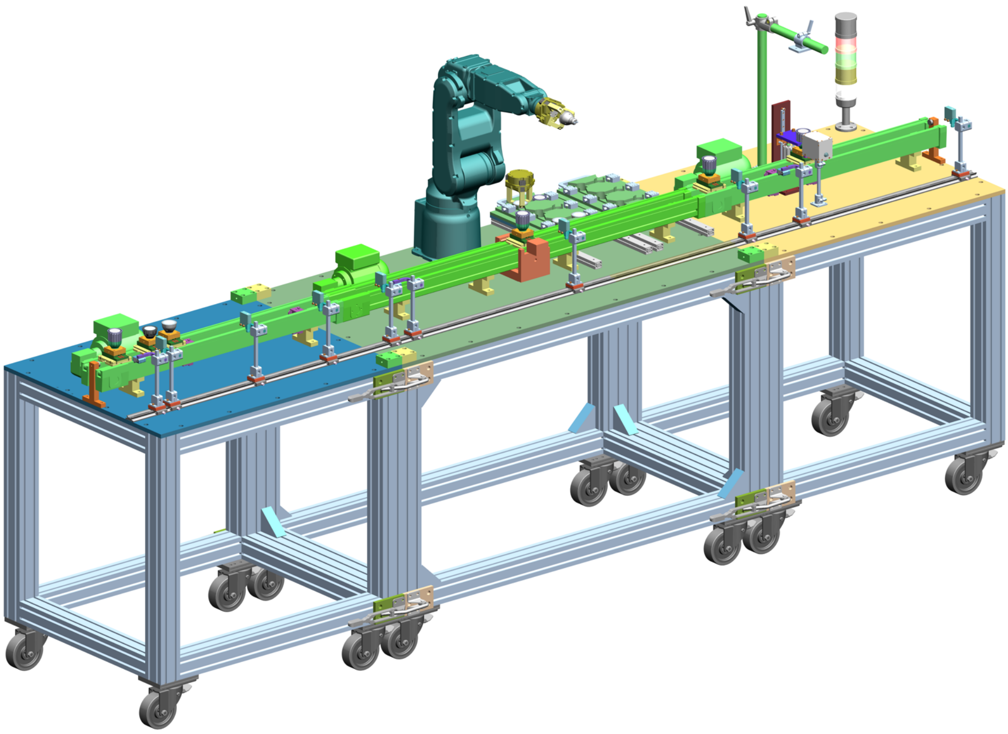

Figure 13 Simplified CAD model of the AKOMI assembly demonstrator.

[1] For example, both a mechanical sensor and also a vision system would offer the identical *Check Presence* skill. The matchmaking software would determine the task-specific suitability.

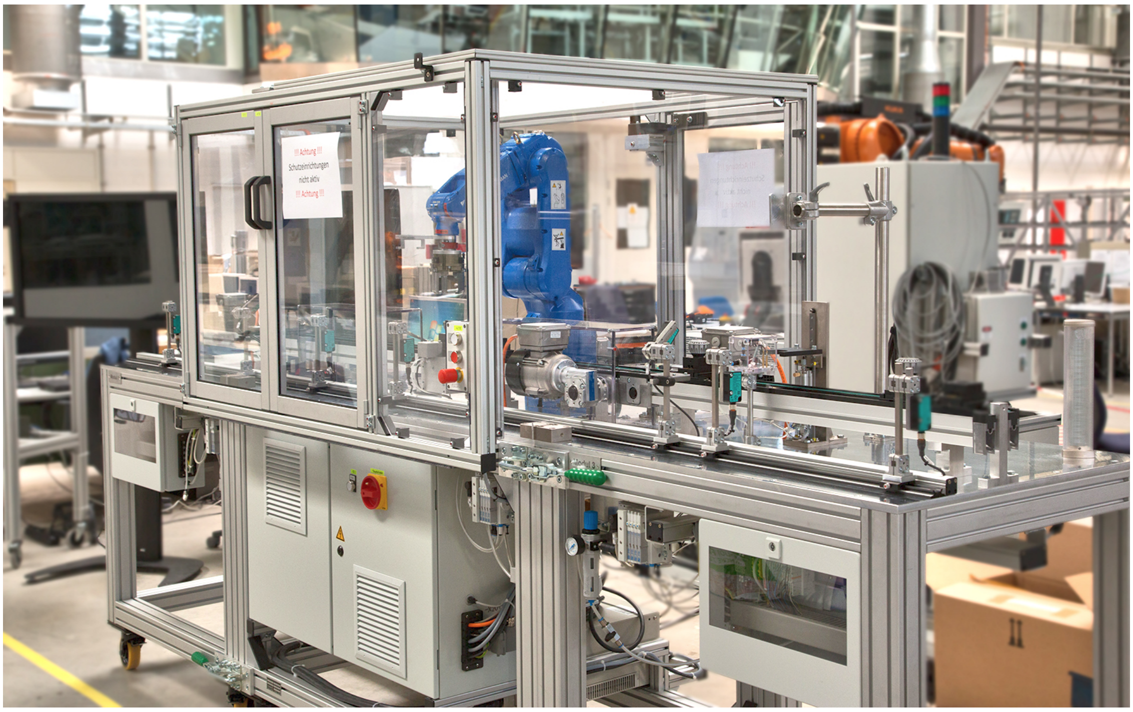

Figure 14 AKOMI assembly demonstrator (Donation from the OSRAM company)

7. Bibliography

Backhaus and Reinhart 2015
Backhaus, J.; Reinhart, G.: Digital description of products, processes and resources for task-oriented programming of assembly systems. Journal of Intelligent Manufacturing (2015).

Backhaus, Hammerstingl et al. 2017
Backhaus, J.; Hammerstingl, V.; Michniewicz, J.; Stocker, C.; Ulrich, M.; Reinhart, G.: Montage 4.0. In: Reinhart, G. (ed.): Handbuch Industrie 4.0. Geschäftsmodelle, Prozesse, Technik. München: Hanser, 2017, pp. 513–554, ISBN: 978-3-446-44642-7.

Cândido and Barata 2007
Cândido, G.; Barata, J.: A Multiagent Control System for Shop Floor Assembly. In: Mařík, V.; Vyatkin, V.; Colombo, A. W. (eds.): Holonic and Multi-Agent Systems for Manufacturing. Vol. 4659. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 293–302, ISBN: 978-3-540-74478-8.

DIN 1319-1
Deutsches Institut für Normung (1995): Grundlagen der Meßtechnik. DIN. Vol. ICS 17.020; 01.040.17. Berlin: Beuth.

DIN 8580
Deutsches Institut für Normung (2003a): Fertigungsverfahren. DIN. Vol. ICS 01.040.25; 25.020.

DIN 8593-4
Deutsches Institut für Normung (2003b): Fertigungsverfahren Fügen. DIN. Vol. ICS 01.040.25; 25.020; 25.120.99.

Feldmann and Spur 2014
Feldmann, K.; Spur, G. (2014): Handbuch Fügen, Handhaben, Montieren. München: Hanser. 2014.

Hammerstingl et al. 2015
Hammerstingl, V.; Moule, L.; Reinhart, G.: Bildverarbeitungssysteme als cyber-physische Sensoren. atp edition 57 (2015) (11), pp. 44–57.

Hammerstingl et al. 2016
Hammerstingl, V.; Reinhart, G.; Zimmermann, P.: Automatisierte Konfiguration und Selb-

stauskunft von Industrierobotern. Intelligente Vernetzung mittels ROS und OPC UA. Industrie 4.0 Management 32 (2016) (6). Eingereicht am 27.11.2015, pp. 17–21.

Hesse and Schnell 2014
Hesse, S.; Schnell, G.: Sensoren für die Prozess- und Fabrikautomation. Wiesbaden: Springer Fachmedien Wiesbaden, 2014. ISBN: 978-3-658-05866-1.

Huckaby and Christensen 2012
Huckaby, J.; Christensen, H. I.: A Taxonomic Framework for Task Modeling and Knowledge Transfer in Manufacturing Robotics. In: Cognitive robotics. Technical report / American Association for Artificial Intelligence WS 2012,6. Palo Alto, Calif.: AAAI Press, 2012, pp. 94–101, ISBN: 978-157735571-7.

Kluge 2011
Kluge, S.: Methodik zur fähigkeitsbasierten Planung modularer Montagesysteme. Fakultät für Konstruktions-, Produktions- und Fahrzeugtechnik. Dissertation. Universität Stuttgart, 2011. Stuttgart.

Loskyll 2013
Loskyll, M.: Entwicklung einer Methodik zur dynamischen kontextbasierten Orchestrierung semantischer Feldgerätefunktionalitäten. Vol. 25 : Kontextadaptive Automatisierung. Fortschritt-Berichte pak. Kaiserslautern: Techn. Univ, 2013. ISBN: 978-3-943995-29-9.

Lotter and Wiendahl 2012
Lotter, B.; Wiendahl, H.-P.: Montage in der industriellen Produktion. Ein Handbuch für die Praxis. 2., Aufl. VDI-Buch. Berlin: Springer Berlin, 2012. ISBN: 978-3-642-29060-2.

Morrow and Khosla 1997
Morrow, J. D.; Khosla, P. K.: Manipulation task primitives for composing robot skills. In: Proceedings of International Conference on Robotics and Automation. International Conference on Robotics and Automation. (Albuquerque, NM, USA). IEEE. 1997, pp. 3354–3359. ISBN: 0-7803-3612-7.

Schmidt 1992
Schmidt, M.: Konzeption und Einsatzplanung flexibel automatisierter Montagesysteme. Vol. Bd. 41. IWB Forschungsberichte. Berlin and New York: Springer, 1992. ISBN: 978-3540550259.

Schmidtke and Jastrzebska-Fraczek 1993
Schmidtke, H.; Jastrzebska-Fraczek, I.: Ergonomie. Daten zur Systemgestaltung und Begriffsbestimmungen. München: Hanser, 1993. ISBN: 978-3-446-43480-6.

Smale and Ratchev 2009
Smale, D.; Ratchev, S.: A Capability Model and Taxonomy for Multiple Assembly System Reconfigurations. IFAC Proceedings Volumes 42 (2009) (4), pp. 1923–1928.

VDI 2860
Verein Deutscher Ingenieure (1990): Montage- und Handhabungstechnik - Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole. VDI-Richtlinie. Vol. 2860. Düsseldorf: VDI.

8. Review

The following persons have reviewed this publication. The reviewers' comments were taken into account accordingly.

- Günther, Till – Member of the Institute Management Committee: Assembly Technology and Robotics, Institute for Machine Tools and Industrial Management (*iwb*) – Technical University of Munich (TUM).
- Leiber, Daria – Research associate in the research project ProMoA (Use of skill-based modelling and matchmaking), Institute for Machine Tools and Industrial Management (*iwb*) – Technical University of Munich (TUM).
- Heidel, Roland – Member of several technical committees and author on the topic "Industry 4.0", roland.heidel@heidelcom.com