2016 IEEE International Conference on Automation Science and Engineering (CASE)
Fort Worth, TX, USA, August 21-24, 2016

Manufacturing Knowledge for Industrial Robot Systems: Review and Synthesis of Model Architecture

Alexander Kuss[1], Julian R. Diaz P.[1], Rebecca Hollmann[2], Thomas Dietz[1], Martin Hägele[1]

Abstract— **This paper reviews the state of the art of models in robotic manufacturing applications. Moreover, a model architecture is proposed that is adequate for capturing knowledge of manufacturing processes in small and medium sized enterprises. The proposed architecture improves existing approaches by introducing a technology model that allows modeling of specific knowledge of manufacturing processes. This enables automatic generation of robot programs and supports reusability of manufacturing knowledge in industrial production scenarios. The usage of the presented model architecture is shown for a robotic welding application.**

I. INTRODUCTION

Manufacturing applications for production in small and medium sized enterprises (SMEs) are characterized by a high number of variants, small lot sizes and frequent changeover. As a consequence, introducing robot systems to their production poses a big challenge to SMEs given the current inflexibility of robot systems. Programming is time-consuming and often too complex as the process experts at the SME are typically not experienced robot programmers. Automatic program generation appears to be a feasible solution. However, for complex manufacturing processes, like welding or deburring, the respective manufacturing knowledge has to be modeled to be used by an automatic programming system. Up to date, this knowledge in SMEs is typically only represented as part of the qualification of the machine operators. Their knowledge should therefore be modeled and considered while planning the process execution. Representing manufacturing knowledge in appropriate models also allows reusability of this knowledge for different products and quality improvement over time e.g. through feedback of a human operator. Fig. 1 shows our robotic welding system including robot, table, workpiece and welding process and the relation to possible model types, like product, process and resource models. To facilitate the modeling of manufacturing knowledge, an adequate model architecture as well as appropriate means for interaction and communication with the robot system are required.

*The research leading to these results has received funding from the European Union Seventh Framework Program (FP7/2007-2013) under grant agreement no. 287787 in the project SMErobotics and European Union's Horizon 2020 research and innovation programme under grant agreement no. 688217 in the project ROBOTT-NET.

[1]Alexander Kuss, Julian R. Diaz P., Thomas Dietz and Martin Hägele are with Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Nobelstrasse 12, 70569 Stuttgart, Germany. All correspondence should be addressed to Alexander Kuss alexander.kuss@ipa.fraunhofer.de

[2]Rebecca Hollmann is with DIgSILENT GmbH, Heinrich-Hertz-Strasse 9, 72810 Gomaringen, Germany.

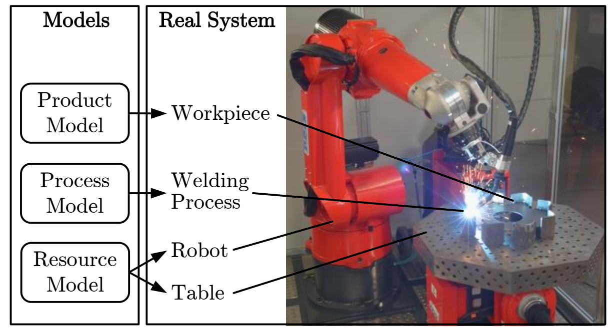

Fig. 1. Models of a robotic welding system

This contribution initially reviews existing model formats in robotic manufacturing applications. Moreover, a model architecture is proposed supporting the modeling of knowledge of specific manufacturing technologies in SMEs. This builds the basis for reuse of this information and builds the basis for the usage of automatic program planning algorithms.

This paper is structured as follows: Section II gives an extensive overview of existing model formulations. Section III describes the proposed model architecture and its implementation details. Section IV demonstrates the usage of the proposed model architecture for a robotic welding scenario. Section V presents the conclusions and gives and outlook for future work.

II. REVIEW OF MODEL FORMATS FOR ROBOTIC MANUFACTURING SYSTEMS

A model is always a simplified representation of a real aspect of a system for a specific purpose. Regarding its intention, different model types can be distinguished. These are e.g. graphic models, technical models, semantic models or cognitive models. To cluster the complex modeling tasks in robot systems, the three view concept, also called PPR-approach, is widespread in manufacturing engineering [1], [2], [3]. It distinguishes product, process and resource models. These three types of models focus on different aspects of the robotic manufacturing system and are all interconnected. In the following subsections we will review existing model formats using the categorization introduced by the PPR-approach.

A. Process Models

In terms of manufacturing engineering the term process typically refers to a sequence of manufacturing operations. In this sense, the process model defines the logic order of

978-1-5090-2409-4/16/$31.00 ©2016 IEEE

different process steps and the connection to all other models, i.e. the used resources and the processed products in each step.

Different formats have been developed to describe processes. *The process specification language (PSL)* is a framework for the description of processes. It was developed with a strong focus on manufacturing processes and is standardized in the ISO 18629. The main aims for its development is to support communication of process data between different software systems and automatic reasoning on processes, e.g. for evaluation of process consistency and compliance to process rules. The logic elements for the process description are defined in an ontology.

RuleML is a knowledge representation for semantic rules in the Extensible Markup Language (*XML*) format and it is defined as an extensible family of sublanguages [4]. One subfamily is *Reaction RuleML* that focuses on production rules and knowledge representation [4], [5].

For the description of manufacturing processes *state chart XML (SCXML)* has been used by different authors [6], [7]. SCXML offers a markup language defined by the World Wide Web Consortium that allows to define complex state-machines [8]. It allows to introduce loops and condition-based branching of the process flow. Several execution engines for SCXML are available. Thomas et. al [9] propose a domain specific language, named *LightRocks*, that is based on UML/P state charts. It is used to generate programs for robotic assembly processes while hiding controller specifics. *Sequential function charts (SFC)* is one of the programming languages for programmable logic controllers standardized in IEC 61131-3 and is commonly used to program high level processes. *SFC* supports branching, simultaneous steps and synchronization of processes.

Additional to the high-level view taken by the PPR-approach, automatic robot program generation also requires information on physical properties of the applied manufacturing processes. This review focuses on a typical manufacturing process in SMEs: metal active gas (MAG) welding. In a MAG welding process the heat of an electric arc is used to melt an electrode wire and the metallic components to be welded [10]. It is mostly used for seam welding, where the motion of the welding tool describes a continuous trajectory in six-dimensional space along the manufacturing workpiece geometry [11]. MAG welding is a complex joining process and research in modeling focuses on different aspects of the manufacturing process rather than on development of standardized model formats.

The development of a heuristic technology model for the welding process for use in robotic applications is described in [12]. A model describing the melting of the filler wire taking into account length variations of the wire is developed in [13]. The usage of a neural network to establish a relation between process parameters and the height of the created weld bead for multi-layer welding is investigated in [14]. A mathematical model based on the assumption that the weld pool is in static equilibrium and that heat transfer to the weld pool is conductive is proposed in [15]. The mathematical modeling of the melting of the filler wire as the most important factor for the productivity of the welding process as well as the influence of the shielding gas on the melting properties is investigated in [16]. A Taguchi design of experiments to determine the parameters of a heuristic model of the MAG welding process for gas pipelines is used in [17]. An approach for automatic identification of welding joints in CAD-data is presented in [18]. The usage of a stereo camera system with laser line projection to detect weld seams and plan the robot path based on the edges of the seam preparation is described in [19].

B. Product Models

Product models describe the properties and geometries of the manufactured goods at different levels of completion. For some technologies, such as machining and welding additional to geometric product information also material properties, material parameters and material behavior are required in the model. For product assemblies also the hierarchical structure of the assembly sequence is required.

The representation of geometry is a pivotal modeling aspect in robotics for the description of resources and products. However, geometric modeling is fragmented, as many competing and incompatible model formats exist. Subsequently, some interesting geometric formats with respect to openness and importance in engineering are presented and further analyzed.

COLLADA is an open file format for 3D data, which is specified in ISO/PAS 17506. It was initiated by Sony and the Khronos Group with the aim to create a standard format for the 3D game industry. *COLLADA* describes 3D geometry under the separated aspects of geometry for rendering, kinematics for articulating the geometry and physics for rigid body simulation. *STEP* is a standardized format (see ISO 10303) for exchange of product manufacturing information. It supports the modeling of data from mechanical and electrical design including tolerancing, machine topology, material properties and process structure. *JT* is a CAD format developed by Siemens and published as standard in ISO 14306. A detailed overview about actual developments in the *STEP* and *JT* formats is presented in [20]. One focus lies is in modeling of product manufacturing information (PMI), e.g. tolerances and surface properties of workpieces. However, up to date these approaches are not sufficient for modeling of complex manufacturing processes, such as welding, and supporting automatic generation of robot programs. *FBX* is a proprietary format for exchange of data from digital content creation tools. *dotXSI* is a text based file format for exchange of geometric information commonly used in architectural design and video games. *VRML* and *X3D* are historically successive formats for describing geometries. Originally *VRML* and *X3D* are designed as 3D formats for the internet. To this end besides geometry, sound and video can also be included.

Product models that consist of different part geometries are typically referred to as assemblies. The ordering of constraints when assembling a product can be modeled by a precedence graph. An approach to model precedence graphs using predicate calculus is presented in [21]. A system that models stereotypical associations between parts of an assembly as assembly features is presented in [22]. The assembly feature models can be detected automatically in assembly CAD models to support automatic generation of robot programs for assembly processes.

C. Resource Models

AutomationML is an open standard specified in IEC 62424 for the exchange of planning data for automation solutions [23]. *AutomationML* files describe the interrelations between different parts of the automation solution, e.g. a gripper as being attached to a robot. Other standards are referenced for different information aspects: *COLLADA* for the description of geometric data and kinematics, *PLCopen XML* for the description of logic interrelations, *CAEX* for the description of the interrelation and component hierarchy of the installation. Due to the use of *PLCopen XML*, only discrete machine behavior can be modeled. *CAEX* is a neutral data format to model the hierarchical connection of different modules and components.

Field device configuration markup language (FDCML), standardized in ISO 15745-3, is a XML based meta language for the description of automation components containing information on device functions, configuration and communication properties of the device as well as the device connections.

Generic station description markup language (GSDML) is a vendor neutral data format storing device properties and configurations for fieldbus communication in particular used for PROFIBUS and PROFInet network configurations.

SysML is a standardized extension of the *Unified Modeling Language (UML)* for modeling of complex systems. It covers the complete development life-cycle from specification to verification and validation and supports modeling the structure of the objects of a machine, behavioral models of the machine (state charts, sequence diagrams, activity diagrams) and requirements management. *SysML* models can be exchanged using the *XML*-based *XMI* standard. *SysML* does not provide facilities for the inclusion of 3D CAD data and, in contrary to *AutomationML*, is not aimed at data exchange between different systems.

D. Conclusions from the State of the Art

A wide range of model formats are available in order to describe manufacturing knowledge in robot systems. Fig. 2 gives an overview of the relevant formats and categorizes them with respect to the information domain of the PPR-approach. It is distinguished whether the model formats are pure information models, describe discrete event systems or represent physical properties.

The models of specific manufacturing processes, required for a physical planning of robot paths, are currently not integrated well with existing model formulations in manufacturing engineering. One reason for this is that research in this discipline is focused on the development of new

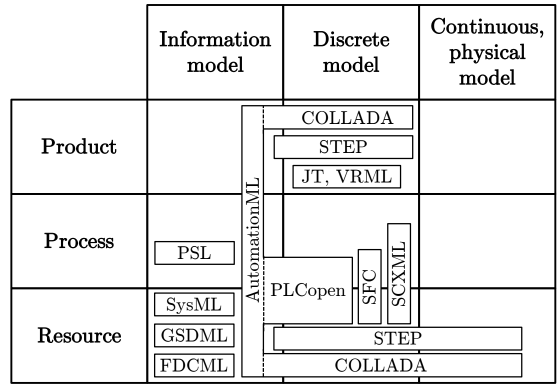

Fig. 2. Comparison of model formats for manufacturing knowledge using the domains of the PPR-Approach

algorithms and not on the formalization of knowledge for usage in automated robot program planning. On the other hand, established model formats, such as *AutomationML*, lack detail with respect to physical models needed in robot program generation. This shortcoming is addressed by this paper by proposing a model architecture that integrates all relevant domains of knowledge to bridge this gap.

III. MODEL ARCHITECTURE SUITABLE FOR SME-LIKE PRODUCTION

For SME-like production a model architecture is proposed that integrates the modeling of specific manufacturing process knowledge. The PPR-approach is therefore extended by a so called technology model. The technology model is closely related to the process model and represents knowledge that is needed to execute a specific manufacturing process, e.g. the required orientation of a welding torch relative to a workpiece. Fig. 3 shows the interrelations of the different model types.

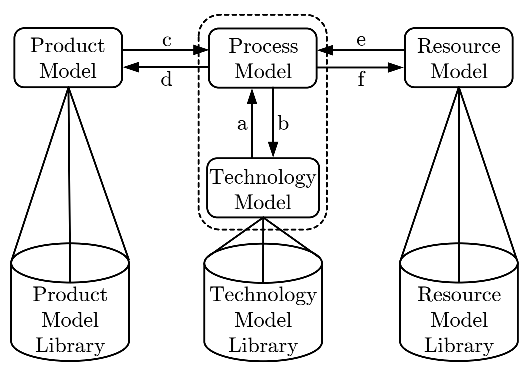

Fig. 3. Overview of different model types and their interrelations

The process model is the basic model which defines the connections to all other models. It represents workflow descriptions of the manufacturing processes to be executed. The process model links to the applicable technology model, which represents the manufacturing knowledge for a specific process (a). The technology models can also be adapted by the process (b), e.g. as a feedback of the process execution.

The product model represents geometric and product related manufacturing information, e.g. the position of a weld seam, that are used by the process (c). Moreover, product related information from different processes steps, e.g. measurement data can be stored in the product model (d). The process model further links to the applied resources models, e.g. the robot and manufacturing tools (e). In case resource parameters change during process execution, e.g. the tool geometry due to collision, this information is also fed back to the resource model (f).

All models could be stored in a respective library, which acts as a knowledge base, but is not focus of this paper. The high-level architecture presented here builds on *AutomationML*. The implementation details for the different models are presented in the following along with examples for welding.

A. Implementation of the Product Model

The implementation of the product model uses the *COLLADA* format as it can be referenced by *AutomationML*. Listing 1 shows an example for a welding workpiece.

```
1   <COLLADA xmlns=”.../COLLADASchema” version=”1.4.1”>
2    <library_geometries>
3     <geometry id=”STATE0”>
4     ...
5     </geometry>
6     ...
7     <extra>
8      <technique profile=”ProductManufacturingInformation”>
9       <url=”../PMI_WeldSeamPlan.xml”>
10     </technique>
11    </extra>
12   </library_geometries>
13   <library_materials>
14   ...
15   </library_materials>
16      ...
17  </COLLADA>
```

Listing 1. Product model of a welding workpiece

The change of the workpiece geometry (e.g from raw product to welded product) throughout the different manufacturing processes is tracked by including different geometry states (see line 3). Material properties are stored in the material library (see line 13) and can be referenced to the respective geometry states. As *COLLADA* does not explicitly support the modeling of product related manufacturing information (PMI) the <extra> extensibility tag is used (see lines 7-11). An example for manufacturing information is a weld seam plan describing information about geometry and position of the weld seams for a specific product model. A more detailed description of the weld seam plan is given in section IV.

B. Implementation of the Resource Model

The natural capabilities of AutomationML are used for the resource models. Fig. 4 gives an overview of the proposed structure for an SME robot cell in AutomationML. Within a robot cell, resources are clustered into four categories: (1) peripherals, either static (e.g. a table or column) or dynamic (e.g. a workpiece positioner). (2) robots including information about its geometry, kinematics and communication interfaces and the end effector configuration. (3) Humans interacting with the system. (4) Manufacturing tools, grippers and sensors, either as determined (tools fixed to the robot) or optional (tools available to the robot, e.g. through a tool changer). For the description of resource geometries *COLLADA* is used.

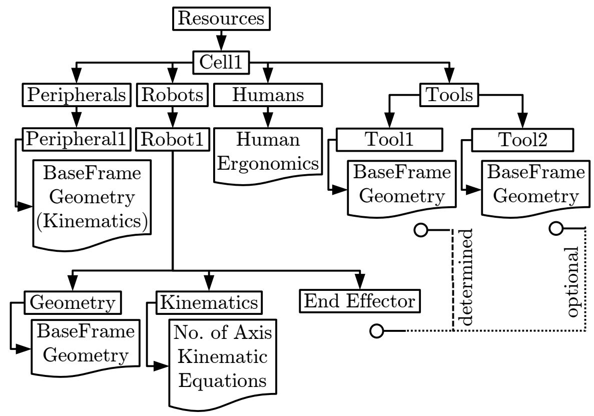

Fig. 4. Architecture of the resource model

C. Implementation of the Process Model

The process model defines the logic order of different process steps and the connection to all other models. The process model is therefore structured in three different levels, as shown in Fig. 5. In the Process Level the process model describes the order of different sub-processes. This high-level process modeling has been discussed in some research, as shown in Section II, and is not the focus of this paper. In the SubProcess-Level the sub-process is defined as a combination of different atomic process models. These atomic process models describe links to an executable software component as well as the connections to corresponding product, technology and resource models.

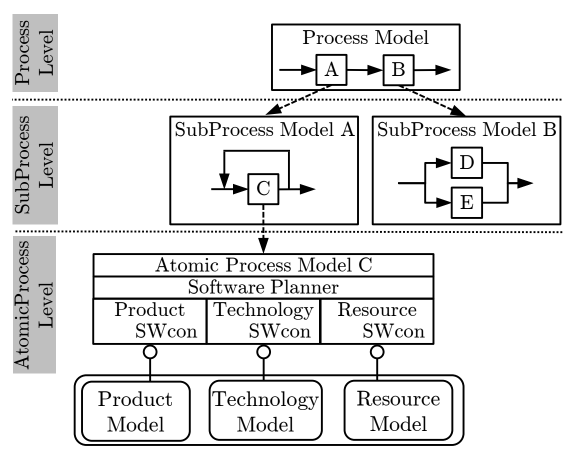

Fig. 5. Architecture of the process model and interconnection to other model types

D. Implementation of the Technology Model

The technology model is implemented in *XML* format. Listing 2 shows a technology model for a MAG welding technology. Firstly, the technology is specified as MAG welding process (see line 1). Moreover, this technology can be categorized by the seam type (e.g. a fillet or butt weld) according to ISO 22553 for welded, brazed and soldered joints. In lines 2 to 3 specific welding parameters (e.g. voltage, ampere, welding speed etc.) are specified. For automatic program planning it is also essential to define how to convert a geometric feature of the workpiece (e.g. an edge) to an adequate position of the robot end effector (see lines 4-13). This information can be represented as a rule (see line 5), e.g. using the *ReactionRuleML* format. The detailed description of rules is out of focus of this paper. Moreover, information about additional tool frame constraints for this technology is represented (see lines 6-12) as three different types: If a tool frame parameter does not allow any adjustment it is marked as fix type. If the tool frame parameter can be adapted in a certain range it is marked as range type (e.g. in welding the rake angle of the welding gun is typically allowed to vary around 5 to 10 degrees). If a tool frame parameter is marked as open, it is not a constraint for the specific technology (e.g. in welding the welding gun can be rotated around the z-axis without any influence on the welding process itself). This additional constraint information can be used in a later motion planning component to optimize the motion of the tool frame (e.g. for varying the z-rotation of the welding gun to smooth the robot motion).

```
1   <Technology name=”MAG Welding” type=”FilletWeld”>
2    <P name=”Voltage” unit=”V” Val=”180”/>
3    ...
4    <GeoFeatureToToolFrame name=”GeoFeatToToolFrame_Fillet”>
5     <Rule url=”../R_Fillet_GeoFeatToToolFrame_Fillet.rrml”>
6     <AdditionalFrameConstraints>
7      <P name=”x” unit=”m” type=”fix”/>
8      ...
9      <P name=”b” unit=”deg” type=”range”
10      min=”−5” max=”5” default=”0”/>
11     <P name=”c” unit=”deg” type=”open”/>
12    <AdditionalFrameConstraints/>
13   </GeoFeatureToToolFrame>
14   ...
15  </Technology>
```

Listing 2. Technology Model in XML format

IV. APPLICATION OF MODEL ARCHITECTURE FOR ROBOTIC WELDING

The usage of the proposed model architecture for robotic manufacturing systems is shown based on a MAG welding use case. Fig. 6 shows the application scenario including robot, welding tool, workpiece and weld seam. The evaluation of the approach focuses on the weld seam detection process as one example of the different planning steps.

A. Program Planning based on Model Architecture

The model architecture can be used by a software component for planning the robot program. The planning phase is meant to be executed in a semi-automatic way, which means that user interactions might be used for certain process steps, if there is no automatic software component available (e.g. for sequencing of weld seams). The goal of the planning phase is the generation of a robot program, that can be executed by the robot to perform the welding process. Fig. 7 shows the concept of the usage of our model architecture in a software for robot program planning.

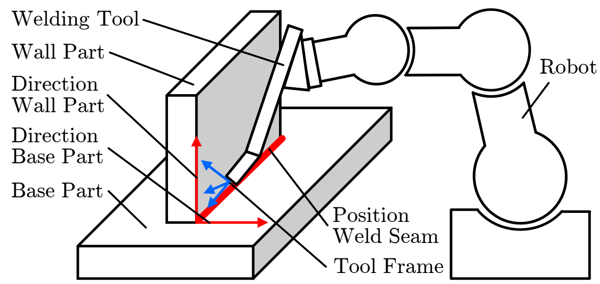

Fig. 6. Concept of robotic welding application

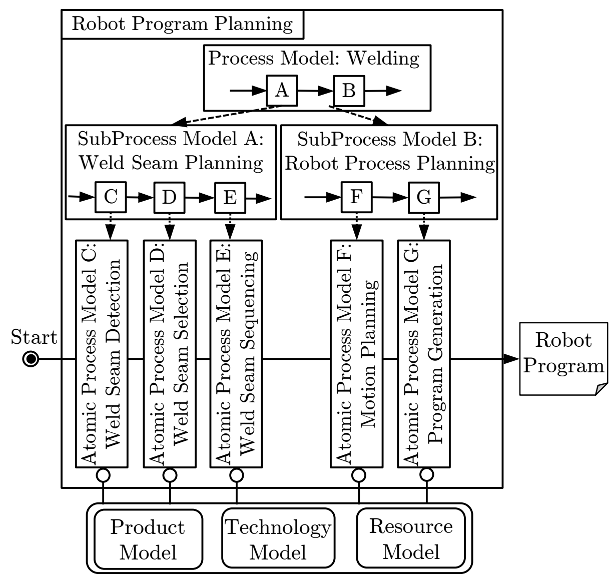

Fig. 7. Application of model architecture for automatic program planning in robotic welding

The process model is used as a basic input for the program planning as it defines the order of different process steps and the connections to all other models and software components. The models and their interrelations can be specified and adapted according to the specific manufacturing processes and knowledge in the SME-production. In our welding example the process model consist of a SubProcess model A for weld seam planning and a SubProcess model B for robot process planning. These sub-processes can consist of a composition of different AtomicProcess models which define links to a corresponding software component as well as to the corresponding product, resource and technology models. The planning based on these two sub-processes and the related models is described in the following, while the description of the particular software components and functionalities is out of focus of this paper.

The purpose of the weld seam planning is to define geometry, parameters and sequence of weld seams for a specific workpiece. This product related manufacturing information is gradually enriched by the software components linked in AtomicProcess models C, D and E and is modeled in a weld seam plan. The weld seam plan is part of the product model as described in section III.

The AtomicProcess model C defines the link to a software component that initially detects possible weld seams on the geometry of a product model based on a geometric feature description of different weld seam types, e.g. fillet or but weld. The weld seam type is defined in the technology model (see line 1 in Listing 2). Listing 3 shows the resulting weld seam plan. In lines 2 to 19 the detected weld seams are specified as <WeldTask> elements. Every weld seam has an ID number and a type, e.g. fillet weld (see line 3). Each weld task is represented by welding contour segments (see lines 4 to 14), describing the edge of a fillet weld as three vectors for position, base direction and wall direction, as indicated in Fig. 6.

In a next step, a welding expert selects the seams to be welded from all detected weld seams. The AtomicProcess model D defines the human input as a resource model and the link to a software with an interactive graphical interface where the user can select, from all detected weld seams, the ones that have to be welded. As a result the respective weld task status is marked active or inactive (see line 3 and 16).

AtomicProcess model E links to a software component, where the process expert can interactively define the desired welding sequence. The result is modeled in the weld seam plan in the <sequence> tag, as a sequence of references to the respective ID of the <WeldTask> elements (see lines 20 to 22). The resulting weld seam plan serves as an input for the subsequent robot action process.

```
1  <WeldSeamPlan>
2   <Tasks>
3    <WeldTask ID="1" type=FilletWeld status=active>
4     <WeldingContourSegment ID="0">
5      <PositionSegment>
6      ...
7      </PositionSegment>
8      <DirectionBaseSegment>
9      ...
10     </DirectionBaseSegment>
11     <DirectionWallSegment>
12     ...
13     </DirectionWallSegment>
14    </WeldingContourSegment>
15   </WeldTask>
16   <WeldTask ID="2" type=FilletWeld status=inactive>
17   ...
18   </WeldTask>
19  </Tasks>
20  <TaskSequence>
21   1 2
22  </TaskSequence>
23 </WeldSeamPlan>
```

Listing 3. Weld seam plan in XML format

The AtomicProcess model F links to a software component that generates a robot motion to move the robot end effector along a workpiece edge without collisions. The basic input of AtomicProcess model F is the weld seam plan (see Listing 3). It defines constraints for the robot motion along the workpiece as well as information about the sequence of weld seams. For transferring the motion constraints to a tool frame position of the welding gun the rule definition in the technology model is used (see line 5 in Listing 2). The technology model also defines possible degrees of freedom as well as possible ranges of parameter deviations (see lines 6 to 12 in Listing 2) that can be used by the motion planning software. To avoid collisions the geometry representations of the product and resource models are also needed as an input for AtomicProcess model F. The motion planning software can then calculate a collision-free robot motion, including the information about mentioned motion constraints. AtomicProcess model G links to the resource model, where information of robot specific programming languages can be described. We assume that AtomicProcess model F and G link to the same software component, which has an internal representation of the motion and logic parameters. This finally allows the generation of a complete program for a specific robot. The robot program can then be executed on the robot controller of the welding system to perform the welding process.

B. Discussion on the efficiency of the proposed approach

Although the proposed approach is applicable to the whole architecture presented in Fig. 7, for the sake of simplicity we only focus on the weld seam detection (see Process C in Fig. 7) as a representative example process in typical robotic welding to discuss the efficiency of the proposed approach.

Minimum typical steps for weld seam detection in traditional offline programming software are as follows [24]: (a) Generation of product CAD model, (b) Extraction of end effector position tags from CAD model, (c) Adaptation of tags (d) Assignment of weld parameters. All these steps typically require user input by a programming expert, who at the same time is an expert in the welding technology. Employment of the proposed approach in this paper reduces the need of user input to a minimum in step (a), while all other steps can be automated, which clearly results in a significant process simplification. The required effort is to once model the respective knowledge used in automatic planning. This effort in each automated step (b), (c) and (d) is detailed as follows.

Step (b) demands for an effort to model knowledge about geometric workpiece features in the technology model. There exist several approaches to be used for automatic weld seam feature extraction, that can be adapted to retrieve knowledge about the geometric features from the technology model [25] [26]. Step (c) requires an effort to describe possible adaptations of tags in the technology model. In welding e.g. where the rake angle of the welding gun can be adjusted up to a certain range, the representation of this degree of freedom in the technology model can be used to smooth the weld path or to compansate for joint position errors [27]. In step (d) specific welding parameters, like e.g. voltage or wire speed, have to be described in the technology model. SMEs typically have very detailed knowledge about their specific welding parameters for different products. This knowledge can be reused to automatically assign welding parameters.

As discussed, the proposed approach in this paper reduces the need of human inputs at least in 3 main steps of the weld seam detection procedure. However, this simplification can be similarly applied to the other 4 processes in Fig. 7 (represented by D to G), which reveals a significant process simplification. Moreover, the proposed approach enables reusability of manufacturing knowledge in other SME productions that use automatic planning components to facilitate the programming of robotic manufacturing tasks.

V. CONCLUSIONS AND FUTURE WORK

We have reviewed the state of the art in modeling of industrial robot systems. While there exist many formats for modeling processes, products and resources in robotic applications, models for representing physical knowledge of manufacturing processes are currently not well integrated with existing model formulations.

This shortcoming was addressed by extending the well known PPR-model architecture by a so called technology model that allows modeling of SME specific manufacturing knowledge. The implementation details and interrelations between process, product, resource and technology models for robotic manufacturing systems have been presented.

The application of our model architecture has been shown for a robotic welding scenario. The representation of specific manufacturing knowledge in appropriate technology models allows the reuse of this knowledge on different manufacturing systems in an industrial production. Moreover, it supports automatic program generation for robotic manufacturing systems. This is essential to establish more robotic manufacturing systems in SME productions, which are characterized by high number of variants, small lot sizes and frequent changeover.

Future work will focus on identification of more complex technology models. Moreover, appropriate software components will be developed that use the model architecture for automatic robot program generation. As the structure of this approach allows the dynamic exchange as well as the extension and adaptation of models, it can also be used as a basis for the application of machine learning algorithms.

REFERENCES

[1] M. Schleipen, and R. Drath, Three-View-Concept for modeling process or manufacturing plants with AutomationML. IEEE Conference on Emerging Technologies & Factory Automation, Mallorca, 2009.

[2] A. F. Cutting-Decelle, R. I. Young, J. J. Michel, R. Grangel, J. Le Cardinal, J. P. Bourey, J. P., ISO 15531 MANDATE: a product-process-resource based approach for managing modularity in production management, Concurrent Engineering, 15(2), 2007, pp. 217-235.

[3] A. Björkelund, H. Bruyninckx, J. Malec, K. Nilsson, P. Nugues, Knowledge for Intelligent Industrial Robots, in AAAI Spring Symposium: Designing Intelligent Robots, 2012.

[4] H. Boley, S. Tabet, and G. Wagner, Design Rationale of RuleML: A Markup Language for Semantic Web Rules, in Proceedings Semantic Web Working Symposium (SWWS01). Stanford University, July/August 2001, pp. 381-401.

[5] A. Paschke, H. Boley, Z. Zhao, K. Teymourian, T. Athan, Reaction RuleML 1.0: Standardized Semantic Reaction Rules, RuleML 2012, 2012.

[6] M. Naumann, and A. Verl, High-level integration of components into a robot cell using semantically annotated state chart descriptions, Proceedings of the 7th German Conference on Robotics, 2012.

[7] G. Veiga, et. Al., Separation of concerns on the orchestration of operations in flexible manufacturing, Assembly Automation, 2012.

[8] State Chart XML (SCXML) Working Draft Published, World Wide Web Consortium, 2012.

[9] U. Thomas, G. Hirzinger, B. Rumpe, C. Schulze, A new skill based robot programming language using UML/P Statecharts, IEEE International Conference on Robotics and Automation (ICRA), 2013.

[10] J. Noberto Pires, A. Loureiro, T. Godinho, P. Ferreira, B. Fernando, J. Morgado, Welding robots, in Robotics and Automation Magazine, IEEE , vol.10, no.2, pp.45-55, June 2003.

[11] J. Angeles, A. Rojas, C. S. Lopez-Cajun, Trajectory Planning in Robotics Continuous-Path Applications, in IEEE Journal of Robotics and Automation, vol. 4, no. 4, Aug. 1988, pp. 380-385.

[12] I. Kim, W.H. Kwon, W.H., and E. Siores, An Investigation of a mathematical model for predicting weld bead geometry, Canadian metallurgy quaterly, number 4, pp. 385-392, 1996.

[13] P. Modenesi, and R. Reis, A model for melting rate phenomena in GMA welding, Journal of Materials Processing Technology, vol. 189, pp. 199-205, 2007.

[14] I.S. Kim, J.S. Son et al., A study on prediction of the bead height in robotic welding using a neural network, Journal of Materials Processing Technology, vol. 130-131, pp. 229-234, 2002.

[15] T. Ohji, Y. Tsuji, F. Miyasaka, T. Yamamoto, Mathematical modeling of metal active gas (MAG) arc welding, Journal of material science technology, Vol. 17, pp. 167-168, 2001.

[16] J. Suban, J., and J. Tusek, Dependence of melting rate in MIG/MAG welding on the type of shielding gas used, Journal of Materials Processing Technology, Vol. 119, pp. 185-192, 2001.

[17] F. Kolahan, and M. Heidari, Modeling and optimzation of MAG welding for gas pipelines using regression analysis and simulated annealing algorithm, Journal of Scientific Industrial Research, Vol. 69, pp. 259-265, 2010.

[18] C. Hillbrand, and G. Frank, Enabling automated programming of welding robots by intelligent identification of weld joints from engineering models, International Conference on Intelligent Engineering Systems (INES), 2012.

[19] L. Hi-xi, S. Yong-hua, W. Guo-rong, Z. Xiao-xi, Automatic teaching of welding robot for 3-dimensional seam based on ant colony optimzation algorithm, Second International Conference on Intelligent Computation Technology and Automation, 2009.

[20] G. Kovcs, D. Kochan, JT Format (ISO 14306) and AP 242 (ISO 10303): The Step to the Next Generation Collaborative Product Creation, in Digital Product and Process Development Systems. Berlin: Springer, 2013.

[21] Y. F. Huang, and C. S. Lee, A framework of knowledge based assembly planning, IEEE International Conference on Robotics and Automation, pp. 599-604, 1991.

[22] A. Dixon and J.J. Shah, Assembly Feature Tutor and Recognition Algorithms Based on Mating Face Pairs, Computer-Aided Design Applications, pp. 319-333, 2010.

[23] R. Drath, Datenaustausch in der Anlagenplanung mit AutomationML, Springer-Verlag, 2010.

[24] Z. Pan, J. Polden, N. Larkin, S. Van Duin, J. Norrish, Recent progress on programming methods for industrial robots, Robotics and Computer-Integrated Manufacturing, 28(2), 87-94, 2012.

[25] B. Babic, N. Nesic, Z. Miljkovic, A review of automated feature recognition with rule-based pattern recognition, Computers in Industry 59.4, 2008, pp. 321-337.

[26] D. A. Sanders, G. Lambert, J. Graham-Jones, G. E. Tewkesbury, S. Onuh, D. Ndzi, C. Ross, C., A robotic welding system using image processing techniques and a CAD model to provide information to a multi-intelligent decision module, Assembly Automation, 30(4), 2010, pp. 323-332.

[27] J. Norberto Pires, T. Godinho, P. Ferreira, CAD interface for automatic robot welding programming, Industrial Robot: An International Journal, 31(1), 2004, pp. 71-76.