ManLang: A Requirements Modeling Language for the Production Planning in Manufacturing

Mert Ozkaya[1] and Gulsah Gokhan Gokcek[2]
[1] *Yeditepe University, Department of Computer Engineering, Istanbul, Turkey*
[2] *Eryaz Software, Istanbul, Turkey*

Keywords: Production Planning, Model-based Engineering, Modeling Language, Multiple-viewpoints Modeling.

Abstract: In manufacturing industries, production planning involves making early decisions about the design and production of a product that focuses on different concerns such as the investment to be made, the product (and its parts) to be manufactured, production line, workstations, and production processes. The decisions and requirements about those concerns tend to be specified using office tools in many manufacturing companies (e.g., MS Excel). However, office tools lead to some issues including *(i)*the lack of support for separation of concerns, graphical editing, precise notation set, and error detection, and *(ii)*inconsistent and incomplete requirements specifications. In this paper, we propose a modeling language called ManLang for specifying the production planning requirements graphically using multiple viewpoints (i.e., investment, product, line, workstation, and process). With ManLang, the different viewpoint models may be specified separately in a way that the viewpoint models can also be traced via the defined relationships between the viewpoints. We support ManLang with a graphical modeling editor, which enables for the multiple-viewpoints modeling and the automated model validation against such properties as model completeness and correctness. We further evaluated the usability of ManLang and its editor via a survey conducted among the practitioners from manufacturing industries.

1 INTRODUCTION

Manufacturing industries aim to produce the high quality products in the least amount of time possible. With the advent of industry 4.0, digitalising the manufacturing engineering (Freddi, 2018; Kroll et al., 2018; Chryssolouris et al., 2009) has gained ever-increasing importance to maximise the business productivity, and therefore, the recent technologies and innovations on the systems and software engineering play key roles in making manufacturing more productive and effective. Indeed, many computer-aided software tools have been developed (e.g., the tools for the process simulations and computer aided manufacturing (CAM), robotic and automation simulation software, etc.), which support the product manufacturing from different perspectives including the design of production systems and processes, robot behaviours, their simulations and implementation.

In this paper, we focus on the production planning of manufacturing engineering, which is one of the most important aspects of manufacturing and need to be handled carefully to achieve the optimal investment for a product while maximising the product quality and reducing the cost and effort needed. To this end, we analysed the manufacturing domain together with the domain experts who work for some of the biggest car and truck manufacturers in Europe (e.g., TOFAS, Ford Otosan, and Volvo), which are the partners of the AITOC ITEA project[1] that our research here is motivated from. As we observed via series of interviews, to specify the requirements for production planning, practitioners need to consider different concerns at different level of details where many different but related data need to be recorded and manipulated and diverse decisions need to be made using those data for optimal production planning. The concerns about production planning may include the variants about the investment to be made (e.g., capacity, annual production, robotic/manual time, etc.), the parts and other details of the product to be manufactured, the production line configurations, the workstation configurations for each line, the production processes to be performed in each station and their configuration, and the resources that

[1] AITOC: https://itea4.org/project/aitoc.html

Ozkaya, M. and Gokcek, G.
ManLang: A Requirements Modeling Language for the Production Planning in Manufacturing.
DOI: 10.5220/0011058400003176
In *Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2022)*, pages 395-404
ISBN: 978-989-758-568-5; ISSN: 2184-4895

need to be used in each process. However, our interviews with the manufacturing practitioners reveals that production planning is not performed in an effective and precise way in manufacturing industries. While the product requirements and related models are typically specified in a standardized and better organized way (both in paper format and/or digital format), those specifications are mainly to do with defining the product functionalities in its final state. Information related to how the product should be manufactured is often missing or not sufficient for process and production planning. The production system requirements that will guide and restrict the choice of production lines and their stations, the production processes involved in each line along with the resources to be used are often in different paper formats and are complex to understand and use.

In our AITOC ITEA project, we conducted a survey among 16 manufacturing practitioners to understand practitioners' experiences with the current tools they use for specifying the production planning requirements. So, we observed that practitioners tend to use the office tools (mostly MS Excel) for specifying the product and production system requirements. One may use Excel to specify the requirements data in the form of rows and columns and execute formulas to perform calculations or Powerpoint to draw some boxes and lines. However, such form of specifications are not so helpful. It is not possible with the office tools to specify the requirements about different concerns (e.g., investment, product, line, station, and process) in a way that is easy to understand, communicate, and trace. Indeed, the requirements for different concerns cannot be separated and specified precisely. Also, one may not use office tools to analyse the requirements specifications for such analysis goals as completeness (i.e., the specification includes all necessary details), consistency (i.e., the requirements are consistent with each other), and correctness (i.e., the requirements are syntactically and semantically correct).

To address the problems discussed above, we consider a model-based solution (Seidewitz, 2003; Whittle et al., 2014), which provides such benefits as the separation of concerns, abstraction, graphical modeling, precise specifications, and tool support for modeling editing and the automated model checking. So, in this paper, we propose a new requirements modeling language called ManLang and its supporting modeling editor. ManLang supports the multiple-viewpoints modeling and enables the production planning requirements to be separated into different viewpoint models that each tackle with a particular concern. The supported viewpoints are to do with *(i)*the investment needed for a product, *(ii)*the product to be manufactured, *(iii)*the production line in which the product is manufactured, *(iv)*the stations employed in the production lines, and *(v)*the production processes operated on each line station. ManLang is essentially intended for the specifications of requirements for diverse manufacturing domains and therefore enables the viewpoint models (and their elements) to vary depending on the domain needs. Indeed, one may specify parameters for the viewpoint models and their modeling elements to make them hold specific types of information defined by the users for the particular domain needs. ManLang defines the relationship between the modeling viewpoints and those relationships are ensured via the modeling editor for the traceability between the viewpoint models. Moreover, the modeling editor further checks for the completeness and correctness of the viewpoint models, where the former is concerned with if the viewpoint models include all the elements that are defined by the language and the latter is concerned with if the viewpoint models satisfy the language definition rules (e.g., the multiplicities and relationships).

In the rest of this paper, we firstly discuss the related work about the general-purpose and domain-specific languages that can be used for the production planning requirements. Later, we introduce the ManLang language in terms of the definitions of the viewpoints. Next, we introduce the editor tool support for ManLang. Lastly, we discuss the preliminary evaluation of the ManLang and its editor tool via a survey conducted among among a set of practitioners.

2 RELATED WORK

The literature includes several modeling approaches that could be used for the modeling of production planning requirements. As aforementioned, while office tools are highly popular, they do not aid in the precise specifications of requirements that can be interpreted in the same way by different stakeholders and processed for analysis purposes. One may alternatively use well-known, general-purpose modeling languages such as SysML (Friedenthal et al., 2008), UML (Rumbaugh et al., 2004), and BPMN (Völzer, 2010). These languages have precise graphical notation sets that could be used for the multiple-viewpoints modeling. Note however that those modeling languages have general-purpose scope and do not offer notation set specific to the manufacturing domain. UML may be extended for the production planning domain with their profiling mechanism. However, with the profiling mechanism, one may not violate the existing syntax and semantics rules for UML and remove any elements. The new language must also be a valid UML model. So, whenever the UML language specification is updated, the extended language needs to be modified too. Also, the extended language may only be used with the modeling tool through which the profiling mechanism is used.

The literature also includes domain-specific modeling languages (DSMLs) for the specifications of manufacturing requirements, e.g., (Vjestica et al., 2020; Weissenberger et al., 2015; Petrasch and Hentschke, 2016; Lütjen and Rippel, 2015; Witsch and Vogel-Heuser, 2012; Jiwei et al., 2008). Those languages offer graphical and domain-specific notation sets for specifying manufacturing requirements. However, the main point of interest herein is centered around the production process flow modeling rather than multiple-viewpoints modeling that encompasses a set of inter-related concerns about production planning including the investment, product, and production line and workstations concerns as is the case with our approach. Vještica et al. (Vjestica et al., 2020) proposed a modeling language for specifying the production processes at two levels of abstractions. That is, first the high-level process flow is specified, which is then detailed with execution-specific details for such tools as simulators and digital twin tools. Weißenberger et al. (Weissenberger et al., 2015; Witsch and Vogel-Heuser, 2012) proposed a modeling language that supports the specifications of manufacturing requirements in terms of three viewpoints: technical system, production process, and MES model. The technical system viewpoint is concerned with the technological structure of the plant in which the product is manufactured. The production process model is concerned with the process flow specification in terms of activities and their transitions. The MES model is concerned with the function definitions in terms of input parameters and output results. Petrasch et al. (Petrasch and Hentschke, 2016) proposed a modeling language based on BPMN that enables to specify graphical production process models with the considerations of different Industry 4.0 aspects. That is, one may specify the process models using such concepts as Cloud App, IoT devices, device data, actuation and sensing tasks, HCI, and mobility which play key roles for the Industry 4.0 applications. Lütjen et al. (Lütjen and Rippel, 2015) proposed a modeling approach called GRAMOSA that promotes the modeling of production processes with the consideration of material and control flows. Jiwei et al. (Jiwei et al., 2008) proposed a formal modeling approach for the production process flows which is based on the petri nets (Brauer et al., 1987) that enables to specify formally analysable dynamic models graphically.

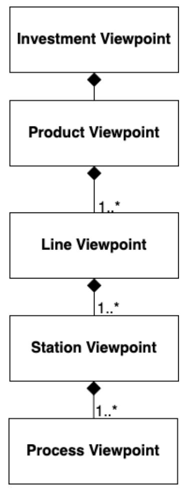

Figure 1: ManLang's viewpoint architecture.

3 ManLang OVERVIEW

ManLang supports multiple modeling viewpoints. These are the investment, product, line, station and process viewpoints. Each viewpoint deals with a different concern of production planning and is supported with a distinct notation set for the specifications of solution models in that viewpoint. In the rest of this section, the viewpoint definitions (i.e., metamodel) are discussed where the viewpoint concepts and their relations are depicted using UML's class diagram notation.

Figure 1 shows that viewpoints are related to each other via the composition (i.e., part-whole) relationship. That is, if a viewpoint is composed of another viewpoint, a model of the latter viewpoint may be accessible via the model of the former viewpoint. Indeed, an investment viewpoint model enables a model of the product viewpoint to be accessible. Also, the viewpoints relate to each other with multiplicities (e.g., 1..*), which constrain the number of the viewpoint models that can be accessible from another viewpoint model. Any product viewpoint model may for instance enable one or more line viewpoint models to be accessible.

Each viewpoint is defined in terms of its metamodel that consists of the concepts, attributes, and their relationships of that viewpoint. A viewpoint concept represents a modeling element for that viewpoint. Any concept may include some attributes that define the data structure of the associated modeling element. Also, a concept may represent another viewpoint. That is, the modeling element for such a concept is essentially a reference to a model of another viewpoint. Concepts may relate to each other via the composition relationship. That is, any modeling element may include some other modeling elements.

Besides the meta-model definitions, each viewpoint is associated with a concrete symbol list that consists of a graphical symbol for each concept of that viewpoint. Likewise, the concept attributes and relationships are also associated with graphical symbols. So, users may use the symbols under certain rules so as to specify the viewpoint models.

The abstract concepts and concrete symbols for each viewpoint have been determined via a series of interviews conducted with the manufacturers contributing to our project[1] and inspired from the previous work conducted in another ITEA project called ENTOC[2] which established the basis for our project.

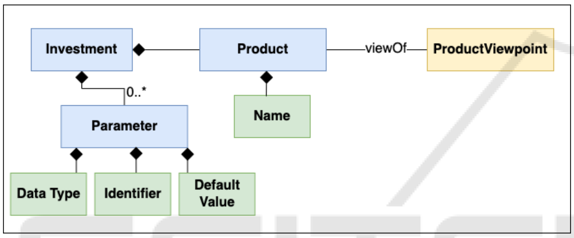

Figure 2: The meta-model definition for the investment viewpoint.

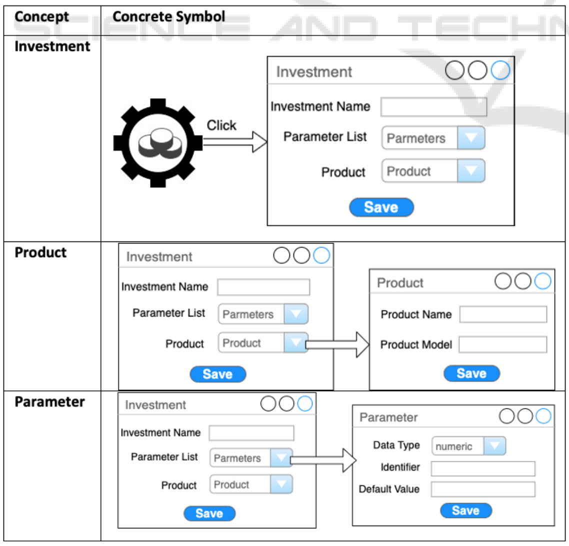

Figure 3: The concrete symbols for the investment viewpoint.

[2]https://itea4.org/project/entoc.html

3.1 Investment Viewpoint

The investment viewpoint is concerned with the investment to be made for any product manufacturing.

Figure 2 shows the meta-model definition of the investment viewpoint, which consists of three concepts – investment, product and parameter. The investment here is the main concept, which is defined in terms of the product and parameter concepts. An investment element derived from the investment concept may consist of zero or more parameters that may be used to specify any data to be used concerning the calculations for a product investment such as the capacity, annual production, working days/hours, robotic/manual time, and engineering cost. A parameter is specified with the data type, data identifier, and value attributes. The investment element may include exactly one product derived from the product concept, which represents the requirements of the product to be invested and is specified with the product name and model instantiated from the product viewpoint.

Figure 3 shows the concrete symbols that correspond to the concepts defined in the meta-model given in Figure 2. The investment concept is mapped to a graphical symbol, and whenever the symbol is clicked, a new dialog box opens for specifying the investment name, product details, and parameters. Clicking product opens up another dialog box for specifying the product name and the model element which is clicked to open up a new sub-editor for specifying the product viewpoint model as discussed in the next sub-section.

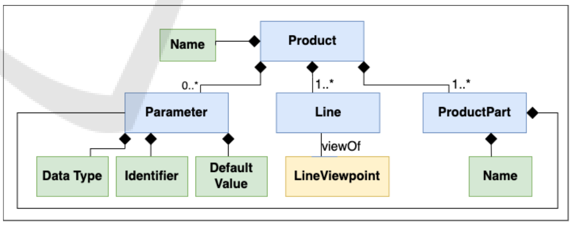

Figure 4: The meta-model definition for the product viewpoint.

3.2 Product Viewpoint

The product viewpoint is concerned with any information required for the product to be manufactured.

Figure 4 shows the meta-model definition of the product viewpoint, which consists of four concepts – product, product part, parameter and line. The product herein is the main concept that represents the product modeling element to be specified. A product element derived from the product concept may consist of the name attribute and the elements derived from

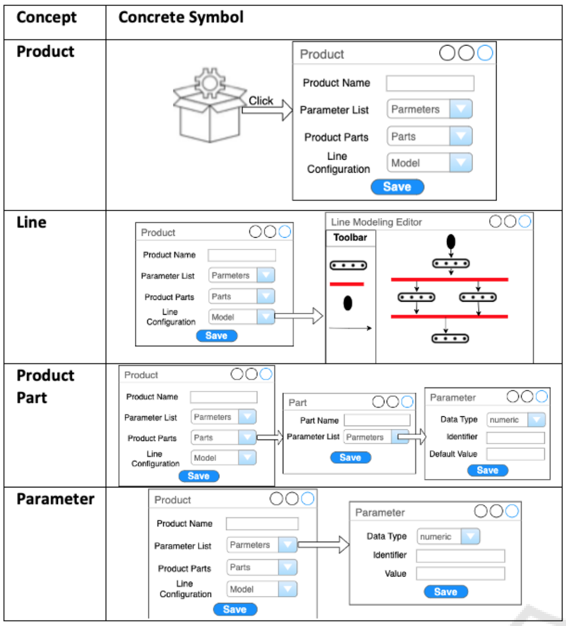

Figure 5: The concrete symbols for the product viewpoint.

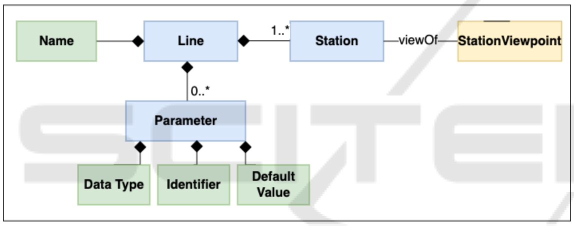

Figure 6: The meta-model definition for the line viewpoint.

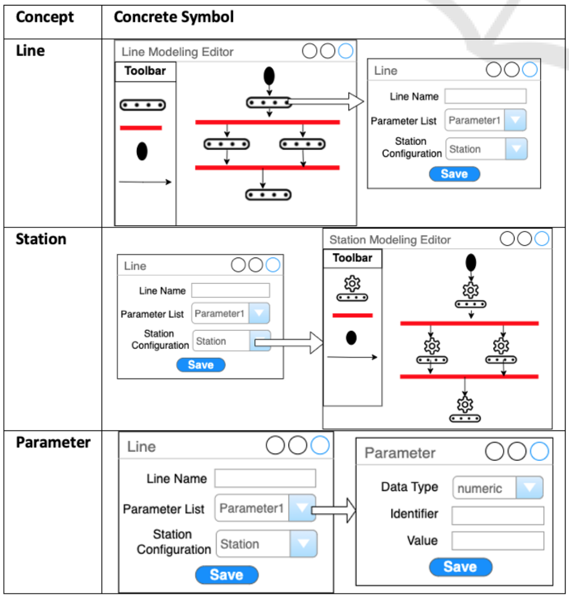

Figure 7: The concrete symbols for the line viewpoint.

the rest of the concepts (i.e., product part, parameter, and line). A product may have zero or more parameters, representing any data about the product to be manufactured and specified in terms of the data type, identifier, and value attributes. A product may have one or more line elements where the product parts are processed to produce the end product. The production of any product may be divided into multiple lines each of which is responsible for a particular set of operations to be performed on the product parts. Multiple production lines may be located sequentially providing input (such as product parts, information, and energy) to each other or in parallel (representing the parallel operations to be performed on different parts). Each line element here is modeled separately as a line viewpoint model. Lastly, a product may be composed of one or more product part elements, each representing a distinct part of the product that could be processed over the line stations and specified in terms of the name attribute and a set of parameters.

Figure 5 shows the concrete symbols for the product viewpoint whose meta-model is given in Figure 4. So, whenever the product model element is clicked in the investment viewpoint model, a new editor opens for specifying the product element with the graphical symbol shown here. Upon clicking the product symbol, a new dialog box opens up for specifying the product details (i.e., parameters, line and product parts). Upon clicking the line configuration on the dialog box, a sub-editor opens for drawing a line configuration model in terms of the line elements and their sequential/concurrent flows. The red line on the editor is employed for indicating the lines that operate concurrently while the black directed arrow indicating the lines operating sequentially.

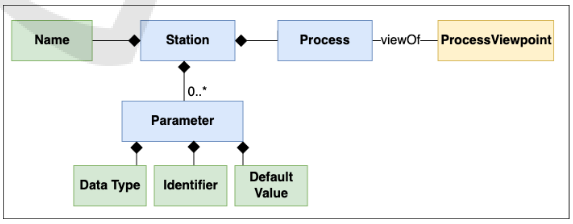

Figure 8: The meta-model definition for the station viewpoint.

3.3 Line Viewpoint

The line viewpoint is concerned with the requirements for a production line where the product parts are processed together with other inputs (e.g., information and energy) via the technical resources.

Figure 6 shows the meta-model definition of the line viewpoint. So, the line element derived from the line concept definition consists of the name attribute

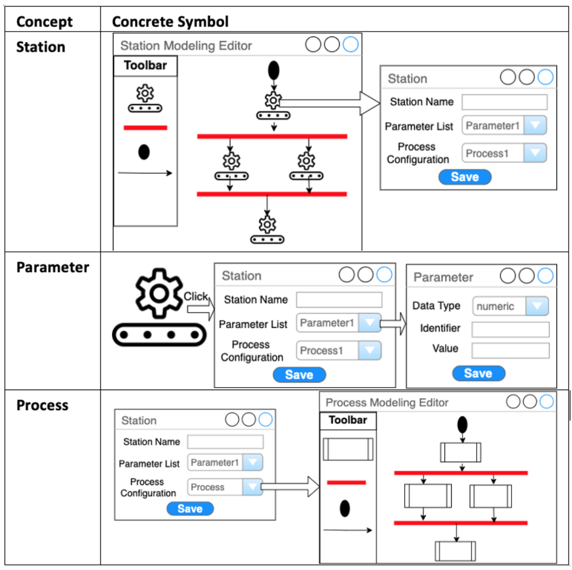

Figure 9: The concrete symbols for the station viewpoint.

and the elements derived from the parameter and station concepts. A line element may be composed of zero or more parameters, representing the data describing the production line (e.g., automation rate of the line). A parameter is specified with the type of data, the parameter identifier and the data value. A line element may also include one or more station elements each of which represents a workstation where a particular set of operation(s) are performed on the product parts. The line stations may be designed to be operated in parallel or sequentially. A station element is modeled separately as a station viewpoint model discussed in the next sub-section.

Figure 7 shows the concrete symbols for the line viewpoint that correspond to the concepts defined in the meta-model given in Figure 6 . So, whenever a line symbol specified in the line configuration model of the product viewpoint is clicked, a new dialog box opens for specifying the line name, parameter list and a station configuration model. Whenever, the station configuration model is clicked on the dialog box, a new sub-editor opens for specifying a station configuration in terms of the stations and their sequential/-concurrent flows. Note that the red line on the editor is employed for indicating the stations that operate concurrently while the black directed arrow indicating the stations operating sequentially.

3.4 Station Viewpoint

The station viewpoint is concerned with the requirements about the workstations employed in a production line where the product parts pass through some processes that are relevant to that line.

Figure 8 shows the meta-model definition of the station viewpoint, which consists of three concepts – station, parameter, and process. Any station derived from the station concept is composed of the name attribute and the elements derived from the process and parameter concepts. Each station element is specified with zero or more parameters for specifying the data about the station. A station element includes one or more processes that perform operations on the product parts with the use of technical resources. Multiple processes for a station may be performed sequentially or in parallel, where the former promotes the processes to provide input to each other (e.g., product parts) and the latter promotes the processes to work on different product parts concurrently.

Figure 9 shows the concrete symbols for the station viewpoint whose meta-model definition is given in Figure 8. Whenever the station symbol is clicked on a station configuration model, a new dialog box opens for specifying the station element in terms of the name of the station, parameters, and the process model. Whenever the process configuration model is clicked, a new sub-editor opens for specifying the configurations of processes involved in the current station in terms of the processes and their sequential/-concurrent flows. Note that the red line on the editor is employed for indicating the processes that operate concurrently while the black directed arrow indicating the processes operating sequentially.

3.5 Process Viewpoint

The process viewpoint is concerned with making decisions about the production processes to be performed in the line workstations. To define a process, we use the VDI 3682 standard[3], which offers a formalised process description. That is, each process is a set of interacting actions in a system that transforms, transports or stores product, energy or information. Any process may accept some input (i.e., product, energy or information) and be performed by a technical resource to produce some output (i.e., product, energy or information). A process is essentially realised by a resource that includes software which run on an embedded device so as to execute the process behaviour.

Figure 10 shows the meta-model definition of the process viewpoint. So, the process viewpoint includes a process concept defined in terms of the input, operation, output, parameter and relationship concepts. That is, a process element is specified with the elements derived from those concepts. Each process is composed of at least one input, one output, and one relationship. An input/output concept may include an

[3]VDI: https://www.vdi.de/en/home/vdi-standards

element which derives from either one of the product part, information, or energy concepts. Each process is composed of one or more operations (e.g., shaping, forming, cutting, joining, and coating). An operation is specified with the name of the operation to be performed and any data parameters about the operation. An operation is linked with a resource to indicate that each process is performed by exactly one resource.

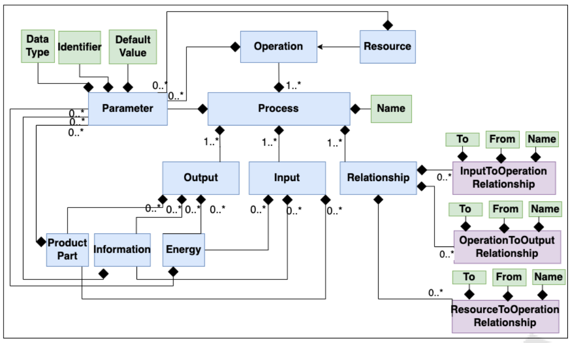

Figure 10: The meta-model definition for the process viewpoint.

A modeling element derived from each concept may possess zero or more parameters to hold some data and any parameter is specified in terms of the data type, identifier and value. A process concept is also defined with the relationship concept definitions. The relationships are the essential part of the process modeling, which compose the operations with the inputs and outputs (i.e., the product parts, information, energy) and also the resources. For any relationship element derived from the relationship definitions, the name, from, and to attributes can be specified, where the from attribute represents the input element and the to attribute represents the output element. The inputToOperation relationship represents the connection from any input elements to the operation, the operationToOutput relationship represents the connection from an operation to any output element, and the resourceToOperation relationship represents the connection from a resource to the operation.

Figure 11 shows the concrete symbols for the process viewpoint that correspond to the concepts defined in the meta-model given in Figure 10. So, whenever a process model element is clicked on the process configuration model that is specified as part of the station viewpoint model, a new dialog box opens. Using the dialog box, the process name and parameter list can be specified. Also, whenever the process model is clicked on the dialog box, a new sub-editor editor opens as shown in Figure 12. With the newly opened editor, the process model can be specified in terms of the elements composing the process.

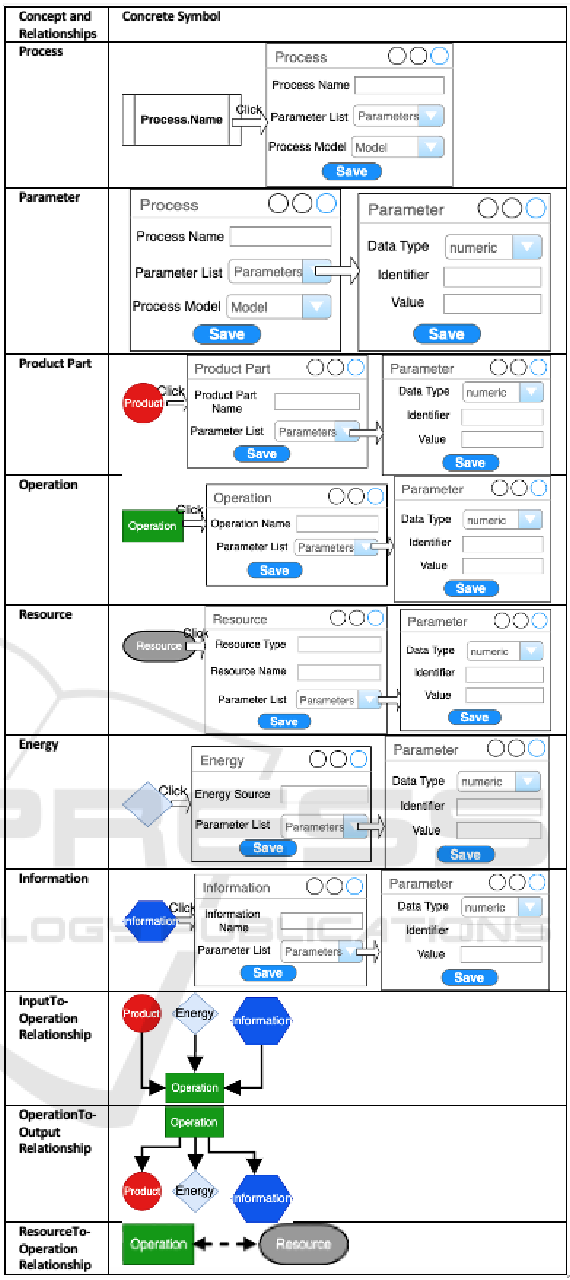

Figure 11: The concrete symbols for the process viewpoint.

4 TOOL SUPPORT

We developed a modeling editor for ManLang using the Metaedit+ meta-modeling tool (Kelly et al., 2013). The editor may be downloaded via the project webpage[4]. We used Metaedit+'s GOPPRR[5] technol-

[4]ManLang: https://sites.google.com/view/manlang

[5]GOPPRR web-site: https://www.metacase.com/support/45/manuals/mwb/Mw-1_1.html

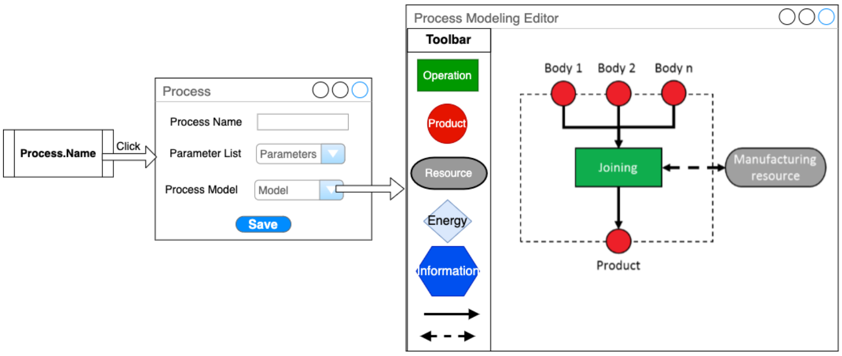

Figure 12: A sample model specified using the process modeling viewpoint notation set.

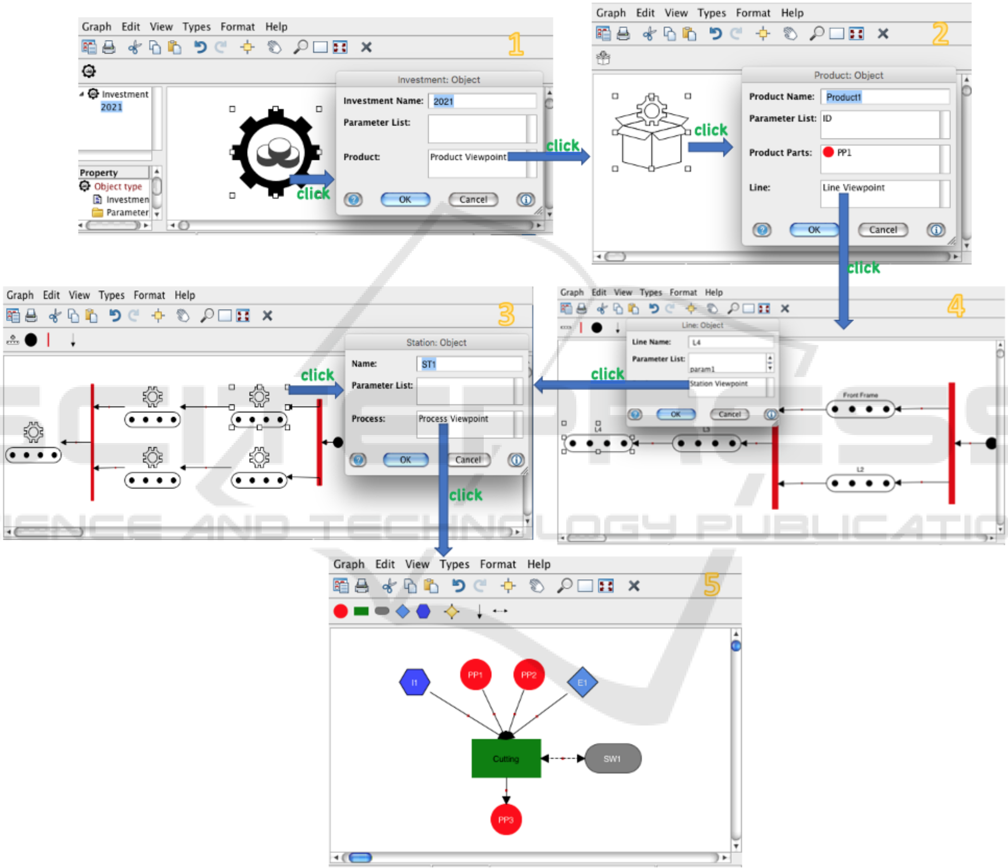

Figure 13: ManLang's modeling editor.

ogy to map the viewpoint definitions given in Section 3 into the Metaedit+ environment. By doing so, we obtained the modeling editor that enables to specify the multiple-viewpoints models in accordance with the ManLang definitions. Figure 13 shows how ManLang's editor can be used to specify the requirements for a production planning, starting with the investment requirements and ending with the process flow model. So, the viewpoint models for any production planning are specified separately but connected to each other and one may traverse different viewpoint models starting from the investment model.

The modeling editor has been extended with an analyser that can check the viewpoint models for completeness and completeness. The editor ensures completeness by checking whether the viewpoint models are specified completely with regard to the notation set of the language viewpoint definitions. For instance, the product viewpoint model that does not have name or any product parts or the process viewpoint model that does not include any resource specification are considered to be incomplete. Such incomplete models are determined by the editor at modeling time and the users are warned with meaningful messages. The editor ensures correctness by checking whether the viewpoint models are correct with regard to the meta-model definitions. That is, each viewpoint model needs to satisfy the multiplicity and relationship rules between the concepts as defined in the viewpoint meta-model. For instance, each station viewpoint model may be composed of a single process model and each investment viewpoint model may be composed of a single product model. If more than one process/product model is tried to be included, this leads to incorrect specifications and thus the editor indicates an error at modeling time.

5 PRELIMINARY EVALUATION

We prepared a usability survey using the well-established usability questionnaire, i.e., System Usability Scale (SUS) (Brooke, 1996). We prepared 10 different Yes/No questions, where participants are expected to choose a single answer. To enhance precision, we designed a scale of answers from 1 (strongly disagree) to 5 (strongly agree). We executed our survey among the participants of our project[1] including the practitioners from such manufacturers as TOFAŞ, Ford Otosan, and Volvo. We initially organised a separate meeting for each manufacturer and introduced our language and its toolset. Then, we asked the participants to fill in our survey that is published online via google form. In total, we got 13 participations for our survey. Some of the interesting survey results are listed below.

- 46% of the participants can frequently use ManLang for the modeling of production planning requirements, while another 46% are unsure and seem to have a lack of knowledge and experience to state their agreement/disagreement.
- 67% of the participants disagree that ManLang's notation set is complex at all.
- 62% of the participants agree that ManLang easy to use for the modeling of production planning requirements.
- 31% of the participants think that technical support (e.g., user manual, tutorials, and help desk) is needed to use ManLang and its toolset, while 30% do not have enough knowledge again.
- 54% of the participants agree that ManLang does not include any inconsistencies about the language definitions and tool support, while 31% do not have enough knowledge.
- 76% of the participants agree that most people would learn to use ManLang very quickly.
- 67% of the participants disagree that ManLang is cumbersome to use.
- 42% of the participants agree that they can confidently use ManLang, while 25% do not have enough knowledge and another 33% disagree.
- 31% of the participants disagree that they need to learn a lot of things before they could get going with ManLang, while another 39% do not have enough knowledge.

It should be noted that we consider the survey results to have some initial idea about industry's thoughts on ManLang. To be able to receive more precise feedback, we are planning to conduct one-to-one interview with each participant where the participant will be asked to spend some time with ManLang and its editor via simple case-studies and share their thoughts over a set of open-ended questions.

6 CONCLUSIONS

In this paper, a new modeling language called ManLang has been proposed for specifying the production planning requirements early at the beginning of manufacturing. ManLang offers a graphical notation set for multiple-viewpoints modeling, which offers a separate (sub-)notation set for 5 different viewpoints (investment, product, production line, workstation, and process). The investment viewpoint is concerned with the information regarding the investment to be made for any product manufacturing, the product viewpoint is concerned with the parts composing the products and any requirements data describing the product and its parts, the line viewpoint is concerned with the configuration of the production lines and their requirements data, the station viewpoint is concerned with the configurations of the workstations for each line and any data about the stations, and lastly the process viewpoint is concerned with the resources involved in each station which perform processes on the product parts, energy, information so as to produce some output. ManLang is further supported with a modeling editor, which enables to specify the viewpoint models separately while establishing the traceability links between them. The editor tool also enables to check the viewpoint models in accordance with the wellformedness rules defined in each viewpoint meta-model.

Our short-term goal is to conduct interview sessions with a group of practitioners to improve the language definitions. We will also develop a model transformation tool using the Metaedit+ technologies so as to transform the ManLang models in accordance with the AutomationML standard (Drath et al., 2008), which is supported by other manufacturing tools and thus enables to process the modeled requirements in ManLang by different tools. We will further extend ManLang for the specifications of resource behaviours (e.g., spot welding robots) and develop a transformation tool for Modelica (Mattsson et al., 1998) which enables to use the simulation tools that support Modelica.

ACKNOWLEDGEMENTS

This work is part of the AITOC (Artificial Intelligence supported Tool Chain in Manufacturing Engineering) project supported by European ITEA and funded by the TUBITAK TEYDEB Project No: 9200078.

REFERENCES

Brauer, W., Reisig, W., and Rozenberg, G., editors (1987). *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part II, Proceedings of an Advanced Course, Bad Honnef, 8.-19. September 1986*, volume 255 of *Lecture Notes in Computer Science*. Springer.

Brooke, J. (1996). SUS: a "quick and dirty" usability scale. *Usability evaluation in industry*, 189(3):189–194.

Chryssolouris, G., Mavrikios, D., Papakostas, N., Mourtzis, D., Michalos, G., and Georgoulias, K. (2009). Digital manufacturing: History, perspectives, and outlook. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 223(5):451–462.

Drath, R., Luder, A., Peschke, J., and Hundt, L. (2008). Automationml - the glue for seamless automation engineering. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 616–623.

Freddi, D. (2018). Digitalisation and employment in manufacturing - pace of the digitalisation process and impact on employment in advanced italian manufacturing companies. *AI Soc.*, 33(3):393–403.

Friedenthal, S., Moore, A., and Steiner, R. (2008). *A Practical Guide to SysML: Systems Modeling Language*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Jiwei, H., Hexu, S., Tao, L., and Zhaoming, L. (2008). The research of manufacturing execution system modeling based on colored petri nets. In *2008 2nd International Symposium on Systems and Control in Aerospace and Astronautics*, pages 1–4.

Kelly, S., Lyytinen, K., and Rossi, M. (2013). Metaedit+ A fully configurable multi-user and multi-tool CASE and CAME environment. In Jr., J. A. B., Krogstie, J., Pastor, O., Pernici, B., Rolland, C., and Sølvberg, A., editors, *Seminal Contributions to Information Systems Engineering, 25 Years of CAiSE*, pages 109–129. Springer.

Kroll, H., Horvat, D., and Jäger, A. (2018). Effects of automatisation and digitalisation on manufacturing companies' production efficiency and innovation performance. Fraunhofer ISI Discussion Papers - Innovation Systems and Policy Analysis 58, Karlsruhe. urn:nbn:de:0011-n-4873361.

Lütjen, M. and Rippel, D. (2015). Gramosa framework for graphical modelling and simulation-based analysis of complex production processes. *The International Journal of Advanced Manufacturing Technology*, 81(1):171–181.

Mattsson, S. E., Elmqvist, H., and Otter, M. (1998). Physical system modeling with modelica. *Control Engineering Practice*, 6(4):501–510.

Petrasch, R. and Hentschke, R. (2016). Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–5.

Rumbaugh, J., Jacobson, I., and Booch, G. (2004). *Unified Modeling Language Reference Manual, The (2Nd Edition)*. Pearson Higher Education.

Seidewitz, E. (2003). What models mean. *IEEE Software*, 20(5):26–32.

Vjestica, M., Dimitrieski, V., Pisaric, M., Kordic, S., Ristic, S., and Lukovic, I. (2020). An application of a DSML in industry 4.0 production processes. In Lalic, B., Majstorovic, V. D., Marjanovic, U., von Cieminski, G., and Romero, D., editors, *Advances in Production Management Systems. The Path to Digital Transformation and Innovation of Production Management Systems - IFIP WG 5.7 International Conference, APMS 2020, Novi Sad, Serbia, August 30 - September 3, 2020, Proceedings, Part I*, volume 591 of *IFIP Advances in Information and Communication Technology*, pages 441–448. Springer.

Völzer, H. (2010). An overview of BPMN 2.0 and its potential use. In Mendling, J., Weidlich, M., and Weske, M., editors, *Business Process Modeling Notation - Second International Workshop, BPMN 2010, Potsdam, Germany, October 13-14, 2010. Proceedings*, volume 67 of *Lecture Notes in Business Information Processing*, pages 14–15. Springer.

Weissenberger, B., Flad, S., Chen, X., Rösch, S., Voigt, T., and Vogel-Heuser, B. (2015). Model driven engineering of manufacturing execution systems using a formal specification. In *20th IEEE Conference on Emerging Technologies & Factory Automation, ETFA 2015, Luxembourg, September 8-11, 2015*, pages 1–8. IEEE.

Whittle, J., Hutchinson, J., and Rouncefield, M. (2014). The state of practice in model-driven engineering. *IEEE Software*, 31(3):79–85.

Witsch, M. and Vogel-Heuser, B. (2012). Towards a formal specification framework for manufacturing execution systems. *IEEE Trans. Ind. Informatics*, 8(2):311–320.