Technical Report

Requirements for product/ion-aware modeling

Lukas Kathrein[1,3]
Arndt Lüder[2]
Kristof Meixner[1,3]
Dietmar Winkler[1,3]
Stefan Biffl[1]

[1]TU Wien, Information Systems Engineering, Information and Software Engineering, Quality Software Engineering, Vienna, Austria
<firstname>.<lastname>@tuwien.ac.at

[2]Otto-v.Guericke Universität Magdeburg, Magdeburg, Germany
<firstname>.<lastname>@ovgu.de

[3] Christian Doppler Laboratory for Security and Quality Improvement in the Production System Life Cycle, Information and Systems Engineering, TU Wien, Vienna, Austria
<firstname>.<lastname>@tuwien.ac.at

Technical Report No. CDL-SQI 2019-03
Issued: March 2019

Citation: L. Kathrein, A. Lüder, K. Meixner, D. Winkler, S. Biffl.: „Requirements for product/ion aware modeling", Technical Report CDL-SQI 2019-03, TU Wien, Vienna, Austria, March 2019.

CHAPTER 5

PPR Modeling Comparison

This chapter addresses the second research issue. The area under investigation of the research issue is how PPR knowledge can be expressed through modeling techniques. All research questions, RQ2a - RQ2c will be answered in this chapter.

The chapter has the following structure: First, a literature survey for existing solutions and requirements will be presented in section 5.1. Section 5.2 discusses then found requirements to be able to model PPR knowledge and commonly found concepts. In section 5.3, the capabilities of existing languages, presented in section 2.4, will be investigated resulting in a benchmarking table for further comparison. Based on the results of existing solutions and requirements, section 5.4 presents adaptations to an existing solution to be capable of expressing PPR knowledge. The chapter closes with section 5.5, an evaluation of the proposed adaptations with a proof of concept implementation and expert interviews regarding the proof of concept.

5.1 Literature Survey

To answer the research questions of this chapter, again an adapted literature survey was executed. The literature survey follows the in section 3.1 presented approach. In this subsection the most vital and differing elements like like: keyword definition, search string definition and the search process execution will be presented here again.

5.1.1 Keyword definition

In this subsection, the used keywords are presented in table 5.1. The key words are simple and atomic strings that can be combined for a search in a digital library as presented in table 3.1.

process	modeling	requirements
taxonomy	production systems	domain
cyber physical systems	comparison	classification
approach		

Table 5.1: Keywords used for the literature survey regarding requirements for modeling approaches.

5.1.2 Search String definition

The search strings are made up of combinations from individual search key words. Not all combinations of the key words do yield a result or would be clever to use, thus only the best combinations are presented in table 5.2.

process modeling	taxonomy of process modeling languages
domain specific modeling languages	modeling for cyber physical systems
modeling production systems	requirements for modeling languages
requirements for modeling languages	comparison of process modeling languages
classification of modeling languages	process modeling language approaches

Table 5.2: Search strings used for the literature survey regarding requirements for modeling approaches.

5.1.3 Search & Initial Analysis

The search strings were executed on the digital libraries, and the resulting titles were examined regarding possible contributions and how they fit the goal of process analysis methods. As the goal of this literature survey is to find requirements and possible approaches to model PPR knowledge, was it not possible to only include the title or abstract. In most papers, the abstract has only very limited results, but for this chapter and research, the requirements are needed which are in most cases an integrated part but not a key outcome. This is why in this step all papers have been scanned for tables presenting requirements, listings comparing languages or headings describing some form of requirements elicitation. Papers, that clearly focused on data models, simulation, time modeling or version modeling were excluded as this is not part of the scope of the language. From this initial analysis, around 45 papers did get selected for further analyses.

Removal of Duplicates and Initial (Quick) Analysis

A criterion which had to be met for a paper to be included further on, was that at least one specific criterion could be extracted or was visible. All papers containing complete tables or listings of criteria were directly included. After this step there were around 15 papers, that could be used to extract requirements out of them, all of these works can be found in an individual bibliography, *Criteria Literature*, at the end of this work, after the regular bibliography.

5.1.4 Data Extraction (Analysis & Results)

In this step all papers were thoroughly investigated and analyzed regarding possible requirements they impose on a modeling language selection. Also, comparisons of different approaches were stripped down into requirement parts so that a requirements catalogue could be built. A lot of work was put into the comparison of different but similar sounding criteria and mapping them to one term. This was done to keep the number of requirements at a minimum and to not completely overload a possible requirements representation. Papers that are focused on code generation or modeling in a non-graphical way like textual descriptions in form of PROSA text were also excluded, as PPR knowledge needs to be represented graphically as motivated in section 1.1. All found criteria and their literature pointers, reasons for inclusion, possible examples and mappings from criteria to PPR modeling requirements will be presented in the next section.

5.2 Requirements for Product, Process, Resource (PPR) modeling

This section presents requirements for modeling PPR knowledge and thus answers the second research question:

"RQ2a: What are the requirements for modeling PPR concepts in a MDE context for PSE?"

Main input for this section is the literature survey from the previous section. But also the design science cycle from [82] is used as a methodological building block. Through interviews with domain experts and stakeholders, was it possible to elicit some requirements and establish a context including goals that should be fulfilled by a possible PPR modeling language. Found requirements from literature were thus presented to the interview partners and they were allowed to prioritize the criterion. The priorities are A for high, B for middle and C for low.

Table 5.3, presents all criteria that were seen worth keeping. Additional requirements, elicited through the interviews, were only added if they were not already part of the existing list from literature. The table is constructed as follows: On the very left side, the first column indicates if the criteria are basic criteria that apply to most common modeling languages or if the criteria belong to a PPR specific group. The second column is an identification number. In column number three, the name of the criterion is then presented followed by the priority of the criterion. Columns five to seven present a mapping of criterion to PPR pragmatics, column eight and nine are requirements for structure or behavior. A row should always be read like this, for example, line one: *"The requirement, relation/flow, has a high priority and needs to be fulfilled from a possible PPR language to be able to express product, process, resource and structural information."*

group	Nr.	criterion	priority	product	process	resource	structure	behaviour
basic elements of modeling language	1	relation/flow	A	X	X	X	X	
	2	logical operators	A		X			X
	3	convergence and divergence	A	X	X		X	
	4	function/activity	A		X			X
	5	result/state	A	X	X	X	X	X
	6	additional parameters	A	X	X	X	X	X
	7	scalability/granularity	B	X	X	X	X	
	8	comments	B				X	
	9	organizational responsibilities	C			X	X	
modeling pragmatics for PPR	10	product assembly modeling	A	X			X	
	11	production process modeling	A		X		X	X
	12	production resource modeling	A			X	X	X
	13	expressing consistencies between PPR elements	A	X	X	X	X	X
	14	parent-child relations	B				X	
	15	relations between PPR concepts	B				X	
	16	relations between the same concept	B	X	X	X	X	
	17	hierarchical structuring of PPR	B	X	X	X	X	

Table 5.3: Criteria for Product, Process, Resource (PPR) language selection.

Each individual requirement, its priority, and mapping are described in more detail in Appendix B. Here only a brief overview will be given of the most important to note elements and facts for a better understanding of the context and later on how the benchmarking results of different languages came to be.

The first group of requirements, the *basic elements of a modeling language*, is characterized by the two-thirds of high priority characteristics. This is easily explained, as the modeling of PPR knowledge, needs to also build on solid common modeling concepts. For example is *relation/flow* crucial to track a path across an engineering process or the execution path of a production system. For more complex systems is it required to be able to express *logical operators*, as definitely not all systems are simple sequential flows. Further, when the modeling of execution processes and their order is of relevance, logical operators are needed to depict details in the execution paths and find possible optimizations through the use of different resource combinations. Other requirements like *function/activity* and *result/state*, make up the main elements of a modeling language and express the very common elements needed, independent of target knowledge which should be expressed. *Convergence and divergence* explain that one process can, for example, have multiple input or output products/resources. Also in the first group of requirements are two entries that are rated medium priority, as it is not of the utmost importance to scale-up the PPR examples, and it is more common to detail them than to use such a language to plan whole factory layouts. *Comments* are also only medium priority, as most knowledge should be represented structurally through the use of *additional parameters*. *Organizational responsibilities* are not rated very important, as little use is seen in using them.

Also, the second group of requirements has more than half of high priority entries. The three most important ones are numbers ten, eleven and twelve, as they build the foundation for PPR modeling and go beyond standard concepts that are present in most languages. All of the three concepts are only then fulfilled by a possible target language if each element can be modeled separately but also in combination. This means that for example, it is possible to model product assemblies without the need for blank or ghost activities that have input/output pins representing the product or object. Additionally to modeling PPR knowledge is it important to express consistencies between these three elements, as currently there are often issues of rework and unstructured communications because some inconsistencies are present. Entries 14 to 17 represent then mainly requirements which are concerned with relationships between the PPR concepts or between the same concept. As seen in section 2.2, the PPR trees are interlinked, and so many relationships are present. Extending this approach, the interviews with domain experts yielded these four individual requirements that should be fulfilled by a possible target language.

As many production systems are focused around the execution of processes and the process view is an important one, is it not surprising that eleven out of the seventeen requirements are needed to express process specific properties. This is only topped by the need to also express structural information. As the modeled information has not only to be visually represented but needs a structured form of representation, is this not surprising. The modeling of product and resource-specific characteristics follows the process modeling and finally, behavioral information has the least amount of mapped criteria.

5.3 Capabilities analysis

This section, the third methodological approach to answer the second research issue, is focused on existing modeling techniques and what their capabilities are. Main outcome is the answer to the third research question:

"RQ2b: What are the capabilities and limitations of different existing modeling language approaches considering the different concepts of PPR modeling?"

The main input for this section is twofold. First relevant and existing modeling approaches from literature and practice serve as a foundation. This foundation is needed to have a baseline of languages that are already used to depict process models and thus provide well-established concepts for modeling PPR knowledge. A second input is the in the previous section presented criteria catalog (see table 5.3). All the found criteria build now a baseline against which the found modeling languages are compared against. The result of this benchmarking is presented in table 5.4 below. The first three columns are the same as previously, grouping, numbering, and name of the criterion. Following this basic frame is a "top" column, which represents the best achievable score if all requirements/criteria would be met. After this benchmark column, the individual found languages are listed and compared against the requirements. The number one (1) indicates that the language fulfills the requirement, whereas the number zero (0) indicates that it is not possible to express the criterion in this specific language. All languages that are now benchmarked have been presented in the related work section 2.4.1.

group	Nr.	criterion	Top	FPD	BPMN 2.0	SySML activity diagram	Petri nets	eEPC	IDEF0	SFC
basic elements of modeling language	1	relation/flow	1	1	1	1	1	1	1	1
	2	logical operators/connectors	1	1	1	1	1	1	0	1
	3	convergence and divergence	1	1	1	1	1	1	1	1
	4	function/activity	1	1	1	1	1	1	1	1
	5	result/state	1	1	1	1	1	1	1	1
	6	additional parameters	1	1	1	0	0	0	0	0
	7	scalability/granularity	1	1	1	1	1	1	1	1
	8	comments	1	0	1	0	0	0	1	0
	9	organizational responsibilities	1	1	1	1	1	1	1	0
modelling pragmatics for PPR	10	Product assembly modeling	1	1	0	0	0	0	0	0
	11	production process modelling	1	1	1	1	1	1	1	1
	12	production resource modelling	1	1	0	0	0	0	0	0
	13	expressing consistency dependencies between PPR elements	1	0	0	0	0	0	0	0
	14	Parent-child relations	1	0	0	0	0	0	0	0
	15	relations between PPR concepts	1	1	0	0	0	0	0	0
	16	relations between the same concept	1	1	1	1	0	0	1	0
	17	hierarchical structuring of PPR	1	1	0	0	0	0	0	0
		sum	17	14	11	9	8	8	8	7

Table 5.4: Benchmarking results for existing languages.

5.3.1 Formal Process Description (FPD)

The FPD fulfills nearly all basic criteria, with the only exception of comments that are not part of the language. It is the only language that actually allows to explicitly model the three different concepts of Product, Process, Resource (PPR) and has for each concept a graphical element. However, there are no capabilities in the language that allows expressing consistency requirements between concepts of PPR knowledge. Further is it not possible to follow parent-child relations which can be seen when processes consist of other sub-processes and the relations are lost or the product modeling on a hierarchical level. All other criteria can be met with the FPD allowing the best result of 14 points out of 17 as a benchmarking result.

5.3.2 Business Process Model and Notation 2.0 (BPMN 2.0)

Focusing on representing business processes, is BPMN 2.0 capable of representing all basic elements of a modeling language. Ranging from relation/flow to functions and comments, all elements are present in this modeling language. Regarding PPR knowledge expression does the approach, however, miss out on crucial parts like explicitly modeling products or resources. Both concepts are not directly representable and do not have their own representations. Further is it not possible to express important relations like parent-child, hierarchical structures or relations between the same concept. BPMN 2.0 achieves a result of 14 out of 17 total points.

5.3.3 Systems Modeling Language (SySML)-Activity Diagrams

The activity diagram extension from SySML, supports many basic requirements. Only additional parameters and comments are missing in the language regarding the basic elements. In regard of the PPR concepts is SySML with Activity Diagram (AD) not very suitable as solution. Neither product nor resource concepts can explicitly be expressed and nearly all other concepts are missing as well. Only the relation between the same concept criterion is fulfilled, as this is a basic concept even to model PPR knowledge. The SySMLAD achieves 9 out of 17 points, resulting in the third place of the overall benchmark.

5.3.4 Petri Nets

Petri nets, as presented in section 2.4, are a perfectly good choice for representing and expressing process flows. However, a major drawback is that the tokens cannot be extended with additional parameters and that no comments can be expressed. An even greater drawback of the mathematically supported modeling approach is, that nearly no PPR requirements are fulfilled. Only processes are depicted, as this concept is also in petri nets a core concept realized as transitions. Petri nets thus achieve only 8 out of 17 points.

5.3.5 extended EPC (eEPC)

The eEPC approach has many similarities with BPMN 2.0, but lacks the possibility to add additional parameters and has no comments integrated. As already stated, is it also a problem of eEPC diagrams that organizational units need to be annotated to every task they belong to, making the diagrams cumbersome to work within larger examples. Also like petri nets before, are eEPC diagrams not suited to express PPR knowledge, as major concepts are missing. eEPC diagrams also score 8 out of 17 total points.

5.3.6 IDEF0

The IDEF0 approach is the only one, that has no explicit concepts for complex logical operators. In the language is it not possible to express AND or XOR branches/joins, making it already hard to work with process language for complex situations. It is further hard to add any additional parameters for example tasks/activities, as the principle is not supported by the standard. IDEF0 has also more drawbacks when looking into PPR modeling. Just like petri nets and eEPC, there are no vital concepts of PPR modeling supported by this approach, achieving 8 out of 17 points.

5.3.7 Sequential Funciton Charts (SFC)

The SFC approach, targeting already the production systems domain with the capability of expressing Programmable Logic Controller (PLC) code via diagrams, performs worst of all investigated languages. Basic elements like additional parameters, comments or organizational responsibilities are missing, making it already hard to express basic PPR concepts. In regard of PPR pragmatics the approach performs worse than all other approaches, making it an impossible task to express PPR knowledge in this language. The approach of using SFC as possible foundation is not well founded, as it only achieves 7 out of 17 points.

Appendix C: PPR Language Criteria Discussion

Result/State

The modeling language is capable of representing results of activities or states which are present at a given point in the model. This could be a certain state of a product f. e. representing the state after a process execution where the two input states become one output state – represented as an assembly group.

Justification
This is a very basic and fundamental element that PPR modeling needs. After a process execution a priori state “ante” is transformed into a “post” state [79]. This also can be true for resources or products, or some other parts like tools which do get transported alongside the product. Further should it be possible to model certain results of processes, that they might have on the product or themselves.

Example
Transfers from input to output states and state changes in product-process-resource
Modelling of a state the product is in after a quality check process

Priority: A
Priority Justification
It is important to have a simple representation of states to allow an in-depth modelling of the states which are f. e. present in the resource or products. To better represent the requirements of a real-world application states/results need to be present and express what has happened because of a task execution.

Mapping Explanation
Result/State modeling is required to be able to fully express PPR concepts and also to be able to represent structural information from the real world including behavioural aspects.

Literature References
[Cri10] [Cri12] [Cri8] [Cri4][Cri2] [Cri14][Cri7][Cri3]

Additional Parameters

VDI/VDE 3682 [79] describes the progress of a procedural modelling approach like this:
1. Graphical representation of the process
2. Information model of the process objects and their connection
3. Attributes of the process objects

This shows that there is a need to be able to attribute elements in a modelling language. For the needs in PPR modelling, this should however not be limited to processes only but should span over all elements including products and processes.

Justification
The roles who are working with the model should be able to represent their knowledge and decisions in some form in the model.

Example
Integer values like acceleration for a process step
(structured) text to make important notes on a resource attribute/setting
External reference to a description of resource behavior
Mathematical functions on how to calculate certain values
Pre- and postconditions to model the execution of a process
If the precondition is not met the process gets not executed and the transport is made to the next module
If the postcondition is not met – a retrial like a second approach of screwing could be possible

Priority: A
Priority Justification
This criterion represents the link between the context and the modelling language. For example a simulation tool wants that each process step has an attribute duration, this functionality thus has to be provided by the modelling language and is essential. But also for criteria which are language specific like the consistency expression between P+P and P+R, they also need parameters which can be checked.

Mapping Explanation
Additional parameters are needed to extend the common representations of PPR elements with additional information. This additional information might also have impacts on structural or behavioural aspects of a model or the underlying context.

Literature References
[Cri10] [Cri7]

Relation/Flow (as epitome of time)

Flow can also be interpreted as execution order. Relation of objects like before/after (similar to the sequence in time) but also the connections/relations of the PPR elements, which product gets processed by which production process and which production resource executes this process.

In the modelling language, the expression A –> B should indicate that A gets executed before B and that B is executed after A has finished. This is the very basic concept of time. Because there are no other time relevant aspects to consider like GANTT [18] or PERT charts [55][22] at least this notion of time or sequence is needed. This also follows the IPA – input, processing, output principle [14].

Justification
All the above presented relations should be expressible in the language. It is needed so that the relationship between objects can be inferred and can be created in the model.

The graphical representation of a model should already allow for a notion of time which is not explicitly expressed. With this criterion the basic notion of time is represented. It is easily possible to find start and end of a process and read an ordering of the process steps and thus extract non explicit knowledge from the model.

Example

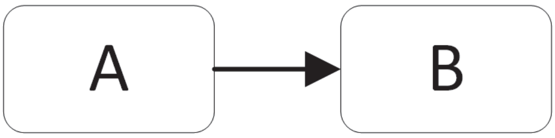

Figure 1: Relation/Flow example where task A is executed before task B is executed.

In the example above (figure 1), there are two tasks: A and B depicted as rectangles with rounded corners. Both tasks are connected via an arrow indicating the relation/flow, meaning that task A is executed *before* task b.

Priority: A
Priority Justification
This represents a very basic concept which needs to be present to be able to model any kind of relationships, due to the fact that the main goal is to model PPR and their relations it is absolutely needed. It is very important to be able to model a flow of process and also assemblies of products etc. This is only possible when a relation/flow element is in place. It is absolutely crucial to have this very basic element of time and sequencing available for the human modeler who creates the model. With this it is possible to express time in a non-explicit way. But also for the interpreter of the model is it of importance to get logical consequences out of a first reading of the model.

Mapping Explanation
The relation/flow criterion is required to be able to express all three of the PPR concepts and additionally structural information of the individual concepts.

Literature References
[Cri8] [Cri4][Cri7][Cri11]

Logical Operators/Connectors

This criterion corresponds to conditions on edges/gateways which allow an alternative routing of the flow and which tasks do get executed. Common operators are parallelism (AND), alternatives (XOR), exception handling triggers and so on. It is a specialization of the requirement “sequential series as epitome of time”, with gateways more complex flows can be expressed.

Justification
I want not only to be able to have sequential flows but also express parallelism and alternatives, thus logical operators like AND/XOR Splits and AND/XOR joins are needed.

Priority: A
Priority Justification
It needs to be possible not only to model sequential flows but more complex flows where operators allow a rerouting or simultaneous execution of tasks.

Mapping Explanation
Logical operators are able to express and manipulate the process flow with resources. But also structural information about the production system and its behaviour can be expressed through this criterion.

Literature References
[Cri10] [Cri13][Cri8] [Cri4][Cri14][Cri7]

Convergence and Divergence

In PPR is should be possible to express the interactions and relationships between the different concepts. Not all of these relations are 1:1, meaning that it is possible that a process has more than one input N:1 or produces more than one output 1:N. The same goes for resources, it is almost all the time the case that more than one resource is needed to completely execute the underlying process.

Justification
The language should not be limited in its expressiveness and allow to model as close to the reality as needed and not impose limitations on the modeler. This is also similar to the “scalability/granularity” criterion there it is more in the direction of aggregation/generalization and here this is continued.

Example
A process requires two input products and welds them together. The process thus has a two to one input to output ratio, also known as convergence.

A process splits one product assembly group up into its individual parts. This is often done if a process is not executed without errors and a (manual) rework disassembles the parts. This is then the divergence concept from one process multiple product parts are outputs.

Priority: A
Priority Justification
Without this requirement the language looses vital expressive features. Only on a very high and abstract level would it be possible to model everything in a 1:1 fashion. But due to the fact, that a PPR modeling language should be able to express more detailed facts it is of high importance.

Mapping Explanation
Convergence and divergence are mainly used to model processes and products, as these two concepts are mostly concerned with this approach. Processes can have multiple inputs or outputs which can either be products or resources. Products are assembled from multiple parts, making up a convergence hierarchy. This kind of information also yields structural knowledge.

Literature References
[Cri7][Cri5]

Function/Activity

Functions/activities represent an executable form of a task. In the concept of PPR this is equivalent to processes which get executed. Tasks do get executed in some order, where here the criterion for "relation/flow" comes into play. This criterion presents a basic building block for many modelling languages and stands in many relations to other elements and criteria to fully enable a language to represent PPR.

Justification
It should be possible to express processes and tasks which get executed either by the resource automatically or by some human worker manually.

Example
Example for tasks are: screwing process, packaging task, but also manual rework tasks or quality checks.

Priority: A
Priority Justification
As stated in the description this criterion makes up a very fundamental part of a language which should represent some form of execution or process modelling.

Mapping Explanation
Processes and their behaviour is mainly expressed by this criterion. All other information is is affected but not directly in contact with this criterion.

Literature References
[Cri1][Cri10] [Cri8] [Cri4][Cri2] [Cri14][Cri7][Cri3][Cri5]

scalability/Granularity

It should be possible to have a higher-level concept of PPR but to be able to go into more depth with each concept. This means, that a process can be a parent process with one product input but in more detail, there are several subprocesses where several input parts of the product assembly are present. This criterion allows for a quick rough layout and first calculation of the most important indicators, and after time allows to detail more fundamental concepts and thus document/model design decisions.

Justification
Different levels of abstraction should be possible to be modeled in a PPR language. This ranges from complete product assemblies with needed processes and resources to lower level concepts of assembly groups.

Example
Model a higher-level process, which can intern have subprocesses. Or model an assembly group for a quicker modeling, and in later engineering phases model the assembly with all individual parts. This can also be visualized like UML Class diagrams with aggregation and generalization [79]

Priority: B
Priority Justification
It is important to have this, but the production process planner often starts with only a rough layout and calculation. This in term means that there won't be much detail anyways in the beginning. Never the less, as time progresses in the project and more details unfold it should be possible to come back to a model and detail it more.

Mapping Explanation
All elements of PPR should be able to contain the same concept as they represent. This allows then to scale models up or down in the level of detail needed. The information about which element contains which other is also structural knowledge.

Literature References
[Cri1][Cri10] [Cri8] [Cri4][Cri14][Cri7][Cri3]

Comments

It should be possible to draw attention to some part of the model which is significant for the context, especially humans in the loop. By adding comments, the model should allow for an obvious way to point to critical knowledge and maybe trace design decisions. The comments should not be used as a form of adding parameters or circumventing the context, like limitations of a tool.

Justification
To make knowledge or certain important notes more explicit and directly visible, there should be an element which allows to model this information.

Priority: B
Priority Justification
It is possible to make a comment in the additional parameter section and not have an extra element for this. But it is a goal of this work to make PPR knowledge explicit and transfer it to later stages in a structured way. If someone else is getting the model and should be ware of special circumstances it is better to display them in the open than to hide them behind parameter settings of objects.

Mapping Explanation
Comments are mainly needed to explain structural or behavioural information. This was a key insight from domain expert interviews and following the design science cycle, that PPR knowledge should be explicit enough to not need any further comments.

Literature References
[Cri10]

Organizational responsibilities

In BPMN 2.0 this is the concept of lanes which also occurs in UML diagrams. This is practical if different roles with responsibilities collaborate to make it explicit which part gets executed under which organizational unit.

Justification
It should be possible to model the responsibilities of modules as an organizational unit. Also in combination with scaling models to higher levels organizational responsibilities gain importance. An example could be the planning of a complete factory.

Priority: C
Priority Justification
In the PPR setting, even if modelling higher level concepts, the organizational responsibility can be identified with the resource or the process or even the product. So, there is already a basic assignment of responsibilities created. It is not of interest to model the organizational responsibilities of different roles such as production process planner. Also is as already discussed the question if for higher level modelling still the same language should be used? Even if so, this criterion should not dominate the basic requirements of the language and the goals.

Mapping Explanation
Modeling organizational units in the context of PPR modeling is only of interest for the responsibilities of resources, as products or processes do not really have a responsibility. However, organizational units are also information conveyors of structural elements.

Literature References
[Cri1][Cri10] [Cri8] [Cri4][Cri14][Cri7][Cri11]

Product Assembly Modeling

Explicitly model products and their assembly via product trees. Here also comes the criterion for hierarchical structuring of PPR into play. It is a specialization of the result/state criterion. Further should this criterion be named product assembly modelling because in PPR it is actually only the case that the assembly is modelled and not the product with its product specific properties like a 3D geometry. Be able to model a product tree all with only product representative information.

Justification It is essential to be able to model the product concept of PPR.

Priority: A Priority Justification
It is essential to be able to model the product concept of PPR.

Mapping Explanation
The modeling of product information is clearly relevant for the product requirement of PPR and also for structural information.

Literature References
[Cri1][Cri12] [Cri8] [Cri14][Cri7][Cri3]

Production Process Modeling

Explicitly model the production processes and their execution. This brings into play the production resource which executes the process and the product which is transformed by the process. It is not the aim to fully model production processes – this means that PPR modelling wants to stay on a higher level than f. e. the engineering process modelling with specific mechanical or electrical process specific models. Be able to build upon the product tree and insert production processes for the assembly groups. Use the products as input for the processes and create outputs from the process execution.

Priority: A
Priority Justification
To be able to model PPR, the process concept is vital.

Mapping Explanation
Process information is needed to model both structural and behavioural aspects of a production system.

Literature References
[Cri8] [Cri2] [Cri3]

Production Resource Modeling

Modeling of the resources needed for specific production processes. Here is it also the aim to model the existence of resources and their part in PPR and not to model physical or geometrical properties. Link production resources to the respective production processes as an individual element in the modelling language.

Justification
To be able to model PPR the resource concept is needed.

Priority: A
Priority Justification
To be able to model PPR the resource concept is needed.

Mapping Explanation
Resource information also contains structural and behavioural aspects of a production system.

Literature References
[Cri8] [Cri2] [Cri7][Cri3]

Expressing Consistencies between PPR Elements

It should be possible to mark links between P+P and P+R as important because there are consistencies which need to be kept in mind.

Justification
Want to make sure that the requirements from a product are met by the process and then also for the resource. Currently there is the problem, that requirements are not fully met or found in the process. It is often the case that the parameters on a resource for a process differ from the actual product requirements.

Example
An example is the screwing process. The product has a requirement, that a force of 30 KN should be applied. The process should know this, that the parameter "force" is 30 KN. If the consistency check is violated there should be an explanation why. The same goes for the selection of a resource. If the process needs 30 KN then all resources are possible where 30 KN is possible. This could then lead to a precondition for resources – needs to be able to screw with > 30 KN

Priority: A
Priority Justification
Currently there is the problem, that often resources are configured differently than the product requirement. In the example above the product needs 30 KN but the robot executes with a force of 42 KN an nobody knows why this decision and derivation was made. This also goes into the direction of traceable design decisions.

Mapping Explanation
This criterion is needed to be able to express the consistencies on a structural level, because the consistencies also form a sort of dependency between two elements. But also are consistencies important to model behavioural aspects, namely what concrete parameters are set for an operational machinery.

Literature References
[Cri9][Cri6]

Parent-child Relations

Expressing the relations between the elements in more detail. Links between the same concept f. e. a higher level process with a nested process is a parent child relation. The link between a process and the production resource would be another relation. Because of these two links between P+P and P+R the relations should be differently modelled. This then allows for a semantic expression of the relations which is not possible with only the relation/flow criterion.

Justification
This criterion needs to be included as it is a part of scaling and nesting different concepts of PPR concepts, as they were already described. For example is it needed to know the relation of parent to subprocess, for a possible execution path.

Priority: A
Justification
PPR does not stand alone all elements in P P R are in a relation like parent child – final product and sub assemblies – so they need relations but also there are relations which will evolve over time and are not categorizable yet.

Mapping Explanation
Parent-child information only contains structural information about the model.

Literature References
[Cri1][Cri10] [Cri13][Cri8] [Cri4][Cri14][Cri7][Cri3]

Relations between PPR concepts

Not only are relations between parent and child objects of interest. Relations between the different concepts of PPR are crucial, as they make up the fundamental parts of the interlinked trees as described in section 2.2.1.

Justification
The criterion needs to be included so that the different concepts can be interlinked. This criterion is to some extend covered and related to *convergence and divergence* but is vital for the PPR modeling, as without it no *convergence and divergence* could be modeled.

Priority: B
Justification
The different concepts need to linked, however it is also possible to express this criterion implicitly, which is why it is only B, priority.

Mapping Explanation
The structuring of PPR concepts is vital for product, process and resource modeling and also contains information about the structure of the model.

Literature References
[Cri13][Cri7]

Relations between the same concepts

As it is a requirement to express consistencies between certain PPR concepts, is it also a criterion to be able to express relations between the same concepts. This ranges from parent-child relations of product assembly groups, to before/after relations of tasks/processes.

Justification
As described above, is this criterion also expressable through other criteria, however it is more important to PPR modeling than to the basic concepts and pragmatics of normal process modeling.

Priority: B
Justification
The relations between the same concepts should explicitly be representable. However, there are already more fundamental criteria which could be used to express this, which is why this is only a B requirement.

Mapping Explanation
The structuring of PPR concepts is vital for product, process and resource modeling and also contains information about the structure of the model.

Literature References
[Cri8] [Cri4][Cri7][Cri11]

Hierarchical Structuring of PPR

When modelling only the product tree a hierarchy is built up naturally, the individual parts are the leaves or children, assembly groups become parents and are also children. The final end product is the root or parent. The same goes for processes, when a process is nested, the highest level represents the parent and all levels below are children to this process.

Justification
Assemblies need to be modeled and structured in a way that the hierarchy they form, from individual part to assembly group to final product can be expressed.

Priority: B
Priority Justification
It is important to be able to express the parent-child relations also in hierarchical form. However this criterion is to some extend already covered by the parent-child concept which is why it is only B.

Mapping Explanation
The structuring of PPR concepts is vital for product, process and resource modeling and also contains information about the structure of the model.

Literature References
[Cri1][Cri10] [Cri13][Cri8] [Cri4][Cri14][Cri7][Cri3]

Criteria Bibliography

[Cri1] Vincenzo Ambriola, Reidar Conradi, and Alfonso Fuggetta. Assessing process-centered software engineering environments. *ACM Trans. Softw. Eng. Methodol.*, 6(3):283–328, July 1997.

[Cri2] M. W. Aziz and M. Rashid. Domain specific modeling language for cyber physical systems. In *2016 International Conference on Information Systems Engineering (ICISE)*, pages 29–33, April 2016.

[Cri3] J. Becker, P. Bergener, D. Breuker, and M. Rackers. Evaluating the expressiveness of domain specific modeling languages using the bunge-wand-weber ontology. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–10, Jan 2010.

[Cri4] R. Bendraou, J. Jezequel, M. Gervais, and X. Blanc. A comparison of six uml-based languages for software process modeling. *IEEE Transactions on Software Engineering*, 36(5):662–675, Sept 2010.

[Cri5] Sébastien Bornot, Ralf Huuck, Ben Lukoschus, and Yassine Lakhnech. Verification of sequential function charts using smv. In *In PDPTA 2000: International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas*. Citeseer, 2000.

[Cri6] Jan Hendrik Hausmann, Reiko Heckel, and Stefan Sauer. Extended model relations with graphical consistency conditions. In *UML 2002 Workshop on Consistency Problems in UML-based Software Development*, pages 61–74, 2002.

[Cri7] Christian Cöllen Hendra. Prozessmodellierung für industrie 4.0 komponenten. Master's thesis, Otto-von-Guericke-Universität Magdeburg, 2018.

[Cri8] Birgit Korherr. *Business process modelling - languages, goals, and variabilities.* PhD thesis, Technische Universität Wien, 2008.

[Cri9] Hugues Malgouyres and Gilles Motet. A uml model consistency verification approach based on meta-modeling formalization. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1804–1809. ACM, 2006.

[Cri10] IR McChesney. Toward a classification scheme for software process modelling approaches. *Information and Software Technology*, 37(7):363 – 374, 1995.

[Cri11] Jan Mendling and Markus Nüttgens. Xml interchange formats for business process management. *Information Systems and e-Business Management*, 4(3):217–220, Jul 2006.

[Cri12] O. Schönherr, J. H. Moss, M. Rehm, and O. Rose. A free simulator for modeling production systems with sysml. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pages 1–12, Dec 2012.

[Cri13] Oliver Schönherr and Oliver Rose. Important components for modeling production systems with sysml. *IIE Annual Conference.Proceedings*, pages 1–6, 2010.

[Cri14] K. Z. Zamli and P. A. Lee. Taxonomy of process modeling languages. In *Proceedings ACS/IEEE International Conference on Computer Systems and Applications*, pages 435–437, June 2001.