Modeling and Verification of Production Process Chains

Jörg Dümmler, Sven Gehre, and Gudula Rünger

***Abstract*—A key element of virtual product development is the modeling of production flows in form of process chains, which consist of processes that are executed either concurrently or consecutively. In practice, process chains can become very complex due to a high number of different processes and, thus, it is difficult for users to maintain an overview of the entire chain. This motivates the necessity to develop models that capture the properties of the individual processes as well as the dependencies between the processes of a process chain on a high level of abstraction. Such models also serve as the basis for various optimization problems, e.g., with respect to the energy usage of the entire process chain. This article proposes a model for the representation of production process chains and additionally shows how to check the consistency of a given process chain model. The modeling is based on a directed acyclic graph with a hierarchical structure. The properties of the processes and the workpieces produced are captured by annotations of the nodes and edges of the graph. The consistency check is based on rules that verify the global structure of the entire process chain, the coupling of processes, and the compatibility of process variants.**

***Index Terms*—Modeling, process chain, production process, verification.**

I. Introduction

Virtual product development uses production process chains to plan and optimize production flows [1], [2]. Such a process chain consists of a sequence of production processes that need to be performed in order to produce a specific workpiece or part of a workpiece. Complex products and workpieces consist of many different parts where the production of each part can be described by a separate process chain. For some of the production processes there may be multiple alternative variants differing in process parameters, resource usage or the type of subprocesses to be performed. For example, a required part can either be produced by a turning process using a big raw workpiece or by welding multiple small raw workpieces. The global optimization for the entire product to be produced, e.g. with the goal to minimize the energy consumption or the number of required processes, has to consider all possible variants for all product parts. Thus, the optimization can become tedious and error-prone when done by hand. This motivates the necessity to provide IT support in form of appropriate software tools.

These software tools have to be based on a suitable model for the representation of production process chains. Such a model has to capture the production processes to be performed with their associated parameters as well as the dependencies between the processes. Currently, there exists a large variety of models to describe business processes (see e.g. [3], [4]) and workflows (see e.g. [5], [6]), but there is no model that is tailored to the specific needs of production process chains.

In this article, we propose such a model, which supports the definition of process parameters such as the welding temperature, the required resources such as energy, and the properties of the input and output workpieces such as the workpiece geometry. The specification of the parameters and resources is flexible, such that a large variety of production processes can be supported and that the extensibility to future developments is ensured. The model also captures the dependencies between the processes, which restrict the possible execution order of the processes, and it also supports different process variants where each variant can consist of an entire process chain. To demonstrate the modeling of a production process chain, we use a real-world example developed in the context of the eniPROD® project [7]. Based on this model, many different optimization tasks can be performed, e.g., with respect to the energy usage of the entire process chain.

To guarantee the consistency of a modeled process chain, it is important to identify possible errors like incompatible input and output of neighboring processes, incompatible process variants, missing processes, or dependencies between processes that prevent a feasible execution order. For this purpose, we propose a set of verification rules that check the global dependence structure, the correct coupling of processes and the correct use of process variants. These rules are formulated in a way that enables the use in software tools.

The article is structured as follows. Section II defines the production process chain model. The modeling for an example process chain is discussed in Section III. Section IV shows how the consistency of a given process chain model can be verified and how the verification can be incorporated in the modeling. Related work is covered in Section V and Section VI concludes the article.

II. Modeling of Process Chains

This section defines a model for production process chains that incorporates both, the specific properties of the production processes and the dependencies between the processes. The model is based on directed graphs, see Subsect. II. A for the basic definitions from graph theory. The global structure of the process chain model is defined in Subsect. II.B, the process coupling in Subsect. II.C, and the modeling of process variants is covered in Subsect. II.D.

A. Basic Definitions

In the following, we define the notations and terms for directed acyclic graphs relevant to this article.

Manuscript received December 10, 2013; revised January 9, 2014.

J. Dümmler, S. Gehre, and G. Rünger are with Technische Universität Chemnitz, Department of Computer Science, 09107 Chemnitz, Germany (e-mail: {djo, svge, ruenger}@cs.tu-chemnitz.de).

Definition 1 (Directed Graph) *A **directed graph** G is a tuple $G = (V, E)$ where*

- *V is a finite set of **nodes** and*
- *$E \subseteq V \times V$ is a set of **directed edges**.*

Definition 2 (Predecessors, Successors, Paths, Cycles) *Let $G = (V, E)$ be a directed* graph.

- *$PRED(v)$ denotes the set of all **predecessors** of a node $v \in V$, i.e., $PRED(v) = \{u \in V \mid (u, v) \in E\}$.*
- *$SUCC(v)$ denotes the set of all **successors** of a node $v \in V$, i.e., $SUCC(v) = \{w \in V \mid (v, w) \in E\}$.*
- *A sequence $(v_1, \dots, v_n)$ of $n \in \mathbb{N}$ nodes is a **path** in graph G, if $(v_i, v_{i+1}) \in E$ for each $i = 1, \dots, n-1$.*
- *A path $(v_1, \dots, v_n)$ is called a **cycle**, if $v_1 = v_n$.*
- *A directed graph without cycles is called **acyclic**.*

B. Modeling of the Process Chain Structure

The dependencies between the production processes forming a process chain are modeled as a directed acyclic graph according to the following definition.

Definition 3 (Process Chain Model) *A production process chain model is a directed, acyclic graph $G = (V, E)$ with the following properties:*

- *The node set V consists of a unique **start node** s without predecessors, a unique **stop node** t without successors and nodes $v \in V \setminus \{s, t\}$ that represent the production processes of the process chain.*
- *The edge set E includes a directed edge $e = (v_1, v_2)$ connecting the processes $v_1, v_2 \in V$ if v_1 produces an output required as an input for v_2. A process $v \in V$ is connected to the start node, i.e. $(s, v) \in E$, if v requires a raw workpiece that is not produced by any other process of the process chain. A process $v \in V$ is connected to the stop node, i.e., $(v, t) \in E$, if v produces a final output of the process chain.*

The process chain model defines the execution order of the production processes. Processes connected by a path in the graph have to be executed one after another. Processes not connected by a path are independent of each other and can be executed concurrently as well as one after another in any order. The properties and parameters of the production processes are annotated at the corresponding graph nodes as explained in the following subsections.

C. Modeling of the Process Input and Output

The input of a production process consists of one or more workpieces, which are further processed and then outputted in a modified form. In the following, we will define a model for a single workpiece first and then consider the annotation of the utilized workpieces in the production chain model.

Definition 4 (Workpiece) *A **workpiece** W is modeled as a triple $(GEO, T, PROPS)$ with the following components:*

- *GEO defines the workpiece **geometry**;*
- *T specifies the workpiece **temperature** and*
- *The **property vector** $PROPS$ is a set of pairs $(prop, val)$ where $prop$ encodes a workpiece property and val defines the corresponding value of the property.*

The specification of the geometry depends on the shape of the workpiece and can, for example, be the cylinder length and diameter of a cylinder-shaped workpiece. Complex geometries can be defined using CAD data [8]. The temperature can either be a scalar value or a feasible range of values.

The additional workpiece properties defined by $PROPS$ include, for example, the workpiece material and the surface conditions. Each property is specified by a pair consisting of an identifier and a value or a range of values. For example, the pair $(ID\_MATERIAL, 20MnCr5)$ defines the workpiece material as steel of grade $20MnCr5$. The pair $(ID\_SURFACE\_ROUGHNESS, [2, \dots, 5])$ defines a feasible range of $[2, \dots, 5]$ for the roughness of the surface.

The workpieces required and produced by a production process are annotated at the corresponding nodes in the process chain model according to the following definition.

Definition 5 (Input and Output Workpieces) *Let $G = (V, E)$ be a process chain model. Each node $v \in V$ is annotated with two sets of workpieces:*

- *The set of **input workpieces** $IN(v)$ includes all workpieces required to perform process v and*
- *The set of **output workpieces** $OUT(v)$ comprises all workpieces produced by process v.*

The raw workpieces of the process chain are annotated as the output of the unique start node s, i.e., as $OUT(s)$, and the final result of the process chain is annotated as the input of the unique stop node t, i.e., as $IN(t)$. Additionally, the edges of the process chain model are annotated with the source and target workpiece as follows.

Definition 6 (Source and Target Workpiece) *Let $G = (V, E)$ be a process chain model. Each edge $e = (v_1, v_2) \in E$ is annotated with two workpieces:*

- *The **source workpiece** $SRC(e)$ is the workpiece produced by process v_1.*
- *The **target workpiece** $TRG(e)$ defines the workpiece required to execute process v_2.*

The source and the target workpiece define the coupling of two processes connected by an edge $e = (v_1, v_2)$. The target workpiece defines all properties of a workpiece that are required to execute process v_2. On the other hand, the source workpiece describes the properties of the workpiece as produced by process v_1. For a correct process coupling, these properties have to match. But some of the properties of the source workpiece may be irrelevant for process v_2, and thus the target workpiece may define a different set of properties than the source workpiece. The target workpiece may also define a feasible range for some properties, whereas the source workpiece defines a specific value for this property. Some properties such as the workpiece temperature may also change due to storage between the two processes.

D. Modeling of Process Variants

Each production process in the process chain model has one or more variants that may differ in the required resources and parameters, e.g., the temperature used in a welding process, or which may consist of a different set of subprocesses. In the following, we discuss the modeling of a single process variant for a given process first and then define the annotation of the process variants in the process chain model.

Definition 7 (Process Variant) *Let $G = (V, E)$ be a process chain model. A **process variant** VAR of a process*

$v \in V \setminus \{s,t\}$ is modeled as triple $VAR = (RES, PARAM, PC)$ with the following components:

- *The* **resource vector** *RES is a set of pairs (res, val) where res defines a resource and val the corresponding quantity of the resource.*
- *The* **parameter vector** *$PARAM$ is a set of pairs $(param, val)$ where $param$ is an adjustable parameter and val the corresponding value.*
- *The* **subordinate process chain** *PC is either NIL (in this case VAR is called* **atomic***) or $PC = G'$ where G' is a process chain model according to Def. 3 (in this case VAR is called* **composed***).*

The resource vector RES of a process variant defines the required resources comprising auxiliaries such as cutting oil or shielding gas, energies like electrical and thermal energy, and byproducts such as slivers and thermal discharge. The individual resources are modeled as pairs consisting of an identifier and a specific value or range of values. For example, the pair $(ID\_COMPRESSED\_AIR, 120bar)$ defines the use of compressed air with a pressure of $120\ bar$.

The parameter vector $PARAM$ of a process variant captures all adjustable parameters like the welding speed or the forging temperature. The definition of a single parameter consists of an identifier and a value or a range of values.

The third element of a process variant is the hierarchically subordinate process chain model PC, which may represent an entire process chain. Composed process variants can be used, for example, if a product consists of multiple parts. In this case, the process on the upper level is the production of a part and the process chain on the lower level defines the production steps required to produce this part. Each composed process variant may include further composed variants. This results in a hierarchical structure of the entire process chain model with atomic process variants on the bottom level of the hierarchy.

Definition 8 (Annotation of Process Variants) *Let $G = (V, E)$ be a process chain model. Each node $v \in V$ is annotated with the set $VARS(v)$ that comprises all variants for process v.*

For a process to be executable, there has to be at least one available process variant. The start and the stop node are not associated with process variants. In the next section, we illustrate the application of the process chain model to a fragment of a real-world production process chain.

III. Example

The modeling of a production process chain is illustrated using a part of the process chain developed in the context of the eniPROD® project [7]. This partial process chain consists of welding round steel to form a shaft consisting of multiple fragments with different diameters and the successive turning of the shaft to produce the required geometry.

The corresponding process chain model $G = (V, E)$ is shown in Fig. 1. It consists of four nodes: the unique start and end nodes s and t, and the process nodes v and w that represent the welding and turning, respectively. For the sake of clarity, we restrict the example to a small number of resources and parameters for the process variants and provide only a selection of the workpiece properties. A real-world process chain model is considerably more complex. The properties, resources and parameters used in the example are summarized in Table I.

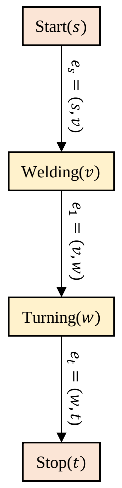


Fig. 1. Nodes and edges of the example process chain model.

TABLE I: Identifiers and Units of the Parameters used in Fig. 2

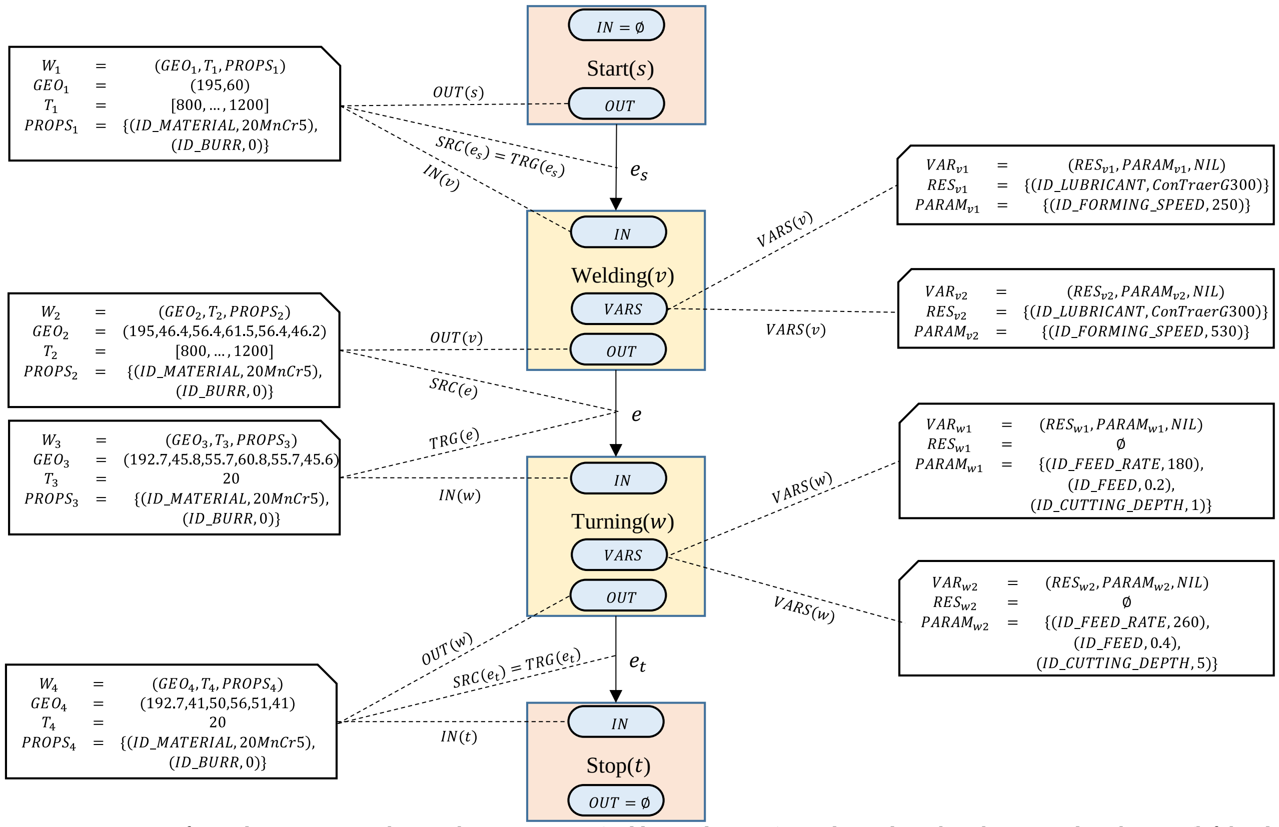
Parameter	Identifier	Unit
Workpiece material	$ID\_MATERIAL$	mm
Burr thickness	ID_BURR	mm
Lubricant	$ID\_LUBRICANT$	-
Forming speed	$ID\_FORMING\_SPEED$	mm/s
Feed rate	ID_FEED_RATE	mm
Feed	ID_FEED	mm
Cutting depth	$ID\_CUTTING\_DEPTH$	mm
Temperature		℃
Geometry		mm

The complete process chain model including the annotations is shown in Fig. 2. The model includes four different workpieces: the round steel W_1, the shaft produced by the welding process W_2, the input shaft of the turning process W_3, and the final shaft after turning W_4. The additional properties specified by $PROPS$, are identical for all four workpieces, i.e., the workpiece material ($ID\_MATERIAL$) is deburred steel of grade $20MnCr5$. Differences exist, however, in the workpiece geometry and temperature.

The round steel used for welding has a length of $195mm$ and a diameter of $60mm$. These extents are specified using the two-element vector $GEO_1 = (195,60)$. The geometry of the shaft after welding is defined using the six-element vector GEO_2, whose first element defines the shaft length and the remaining five elements define the diameters of the five shaft segments. The extents of the shaft change between welding and turning because the workpiece has been cooled down from a temperature between 800℃ and 1200℃ to 20℃. The turning process once again changes the extents of the shaft to the final values of the process chain.

Each of the two processes has two variants. The welding process v has two atomic variants VAR_{v1} and VAR_{v2}, that differ in the forming speed. The first variant uses a forming speed of $250\frac{mm}{s}$ whereas the second variant uses $530\frac{mm}{s}$. Both variants require lubricant of type $ConTraerG300$. The two atomic variants of the turning process (VAR_{w1} and

VAR_{w2}) differ in the feed rate, the feed, and the cutting depth. Both variants produce an identical output workpiece, which is the final result of the example process chain, i.e., is an input workpiece of the stop node.



IN = ∅
Start(s)
OUT
OUT(s)
SRC(e_s) = TRG(e_s)
IN(v)
e_s
IN
Welding(v)
VARS
OUT
OUT(v)
SRC(e)
e
TRG(e)
IN
IN(w)
Turning(w)
VARS
OUT
OUT(w)
SRC(e_t) = TRG(e_t)
e_t
IN
IN(t)
Stop(t)
OUT = ∅
VARS(v)
VARS(v)
VARS(w)
VARS(w)
W_1 = (GEO_1, T_1, PROPS_1)
GEO_1 = (195,60)
T_1 = [800, …, 1200]
PROPS_1 = {(ID_MATERIAL, 20MnCr5), (ID_BURR, 0)}
W_2 = (GEO_2, T_2, PROPS_2)
GEO_2 = (195,46.4,56.4,61.5,56.4,46.2)
T_2 = [800, …, 1200]
PROPS_2 = {(ID_MATERIAL, 20MnCr5), (ID_BURR, 0)}
W_3 = (GEO_3, T_3, PROPS_3)
GEO_3 = (192.7,45.8,55.7,60.8,55.7,45.6)
T_3 = 20
PROPS_3 = {(ID_MATERIAL, 20MnCr5), (ID_BURR, 0)}
W_4 = (GEO_4, T_4, PROPS_4)
GEO_4 = (192.7,41,50,56,51,41)
T_4 = 20
PROPS_4 = {(ID_MATERIAL, 20MnCr5), (ID_BURR, 0)}
VAR_v1 = (RES_v1, PARAM_v1, NIL)
RES_v1 = {(ID_LUBRICANT, ConTraerG300)}
PARAM_v1 = {(ID_FORMING_SPEED, 250)}
VAR_v2 = (RES_v2, PARAM_v2, NIL)
RES_v2 = {(ID_LUBRICANT, ConTraerG300)}
PARAM_v2 = {(ID_FORMING_SPEED, 530)}
VAR_w1 = (RES_w1, PARAM_w1, NIL)
RES_w1 = ∅
PARAM_w1 = {(ID_FEED_RATE, 180), (ID_FEED, 0.2), (ID_CUTTING_DEPTH, 1)}
VAR_w2 = (RES_w2, PARAM_w2, NIL)
RES_w2 = ∅
PARAM_w2 = {(ID_FEED_RATE, 260), (ID_FEED, 0.4), (ID_CUTTING_DEPTH, 5)}

Fig. 2. Representation of a production process chain with two processes (welding and turning) as a directed graph with annotated workpieces (left hand side) and process variants (right hand side).

IV. Verification of a Process Chain Model

In this section, we define a set of rules to verify the consistency of a given process chain model. These rules belong to three categories: rules to check the consistency of the global structure, rules to check the consistency of the process coupling and rules to check the consistency of the process variants. In the following, we discuss these rules and show how to incorporate a consistency check in the modeling of production process chains.

A. Verification of the Global Structure

The following rules ensure a feasible global structure of a process chain model according to Def. 3.

Definition 9 (Consistent Global Structure) *A process chain model $G = (V, E)$ is said to be* ***consistent regarding the global structure****, iff. the following five rules are fulfilled:*

1) *(Start node)*
   The start node $s \in V$ has no predecessor, i.e., there is no $v \in V$ with $(v, s) \in E$.
2) *(End node)*
   The end node $t \in V$ has no successor, i.e., there is no $v \in V$ with $(t, v) \in E$.
3) *(Weak connectivity)*
   Each process node $v \in V \backslash \{s, t\}$ belongs to at least one path from the start node s to the end node t.
4) *(No cycles)*
   The graph G is acyclic.
5) *(Hierarchy)*
   Each process variant $VAR = (RES, PARAM, PC) \in \bigcup_{v \in V} VARS(v)$ is either atomic, i.e., $PC = NIL$, or PC is a process chain with a consistent global structure.

The rules stated in Def. 9 imply the following:

- Rules (1) and (2) guarantee that there is a unique entry and a unique exit point of the process chain at that the raw workpieces for the process chain need to be provided or the final result of the process chain is eventually available, respectively. These points are also required to check the consistency of composed process variants, see Def. 11.
- Rule (3) ensures that there are no orphaned process nodes, i.e., nodes that do not need an input from the start node or nodes that do not produce a part of the final result of the process chain.
- Rule (4) guarantees that there is at least one possible execution order of the processes, which does not violate a dependence defined by the graph edges. Such an execution order can be determined by using a topological sort of the graph G [9].
- Rule (5) is responsible for verifying the consistency of the entire hierarchy of the model, i.e., the consistency of all subordinate process chains defined by the composed process variants.

B. Verification of the Process Coupling

The following rules ensure that the output and the input of two processes connected by an edge are compatible. Using these rules, it is possible to identify missing processes in a process chain.

Definition 10 (Consistent Process Coupling) *A process chain model $G = (V, E)$ is said to be* ***consistent regarding the process coupling****, iff. the following four rules are fulfilled:*

1) *(Process input)*
   Each node $v \in V \setminus \{s\}$ fulfills the inequality

$$IN(v) \subseteq \bigcup_{u \in PRED(v)} TRG\left(e = (u, v)\right).$$

2) *(Process output)*
   For each node $v \in V \setminus \{t\}$, it is

$$OUT(v) \supseteq \bigcup_{w \in SUCC(v)} SRC(e = (v, w)).$$

3) *(Workpiece properties)*
   Each edge $e \in E$ with source workpiece $SRC(e) = (GEO_1, T_1, PROPS_1)$ and target workpiece $TRG(e) = (GEO_2, T_2, PROPS_2)$ fulfills the following two conditions:

a) *The workpiece geometry changes depending on the temperature change and the coefficient of thermal expansion α_c of the workpiece material. In case the geometry and temperature are specified by scalar values, the following holds:*

$$GEO_2 = GEO_1 \cdot \left(1 + \alpha_c \cdot (T_2 - T_1)\right).$$

b) *The temperature of the target workpiece is in-between the temperature of the source workpiece and the temperature of the environment T_{env}, i.e., for scalar temperature values, either $T_1 \leq T_2 \leq T_{env}$ or $T_1 \geq T_2 \geq T_{env}$ holds.*

4) *(Hierarchy)*
   Each process variant $VAR = (RES, PARAM, PC)$ $\in \bigcup_{v \in V} VARS(v)$ is either atomic, i.e., $PC = NIL$, or PC is a process chain model that is consistent regarding the process coupling.

The rules of Def. 10 imply the following:

- Rules (1) and (2) ensure that all input workpieces of a given process are produced by predecessor processes and that each output workpiece of a process is used by at most one successor process, respectively.
- Rule (3) guarantees that the workpiece properties of two processes connected by an edge are compatible. Since the specification of these properties is flexible (see Def. 4), only rules regarding the geometry and the temperature are included. Further rules have to be defined as part of the modeling, see Sect. IV.D.
- Rule (4) is responsible for the verification of the entire hierarchy of process chains, i.e., all subordinate process chains defined by composed process variants.

C. Verification of the Process Variants

The following rules ensure that all variants of the same process require an identical input and produce an identical output. These rules identify incompatible process variants.

Definition 11 (Consistent Process Hierarchy) *A process chain model $G = (V, E)$ is said to be* ***consistent regarding the process hierarchy****, iff. the following three rules are fulfilled:*

1) *(Availability of process variants)*
   For the start node s and the end node t it is $VARS(s) = VARS(t) = \emptyset$. For each process node $v \in V \setminus \{s, t\}$ it is $VARS(v) \neq \emptyset$.
2) *(Variant input and output)*
   For each process node $v \in V \setminus \{s, t\}$ and each composed process variant $VAR = (RES, PARAM, PC) \in VARS(v)$ with $PC = G'$, the following holds: $IN(v) = OUT(s')$ and $OUT(v) = IN(t')$ where s' and t' are the start node and the end node of the process chain model G', respectively.
3) *(Hierarchy)*
   Each process variant $VAR = (RES, PARAM, PC)$ $\in \bigcup_{v \in V} VARS(v)$ is either atomic, i.e., $PC = NIL$, or PC is a process chain model that is consistent regarding the process hierarchy.

The rules of Def. 11 imply the following:

- Rule (1) ensures that each process is executable, i.e., there is at least one available process variant.
- Rule (2) checks the compatibility of the composed variants with the corresponding process v. Each composed variant has to use the same input workpieces as v and produce the same output as v.
  Thus, this rule ensures the consistency of process chain models on different levels of the hierarchy.
- Rule (3) guarantees that the first two rules are followed on all levels of the hierarchy.

D. Modeling and Verification Steps

This section describes the necessary steps to create a process chain model and its verification from the user's perspective. In practice, a specific process may be used in multiple different process chains. Therefore, IT support is recommended for the storage and management of modeled processes. Such a collection can provide a set of standard processes that can be extended and customized. This approach reduces both the effort and the error rate of modeling.

Fig. 3 gives an overview of the modeling and verification steps. In the following the individual steps are described in detail. The creation of a process chain model begins with the specification of processes and their parameters. This includes the corresponding variants of the processes and the number of input and output workpieces. The properties required by the process chain model (for example the resource vector and the parameter vector of a process variant $VAR = (RES, PARAM, PK)$ from Def. 10) initially only contain the corresponding identifiers. The appropriate values or ranges of values will be added later.

Next, the dependencies between the processes are determined based on the input and output workpieces. The dependencies are modeled using a directed acyclic graphm as explained in Subsect. II.B.

Based on the process parameters and graph structure, additional verification rules for the process chain considered are formulated. These rules complete the general rules of Subsections IV.A, IV.B and IV.C and provide an additional consistency check. For example such an additional rule can check, whether a given value is within the allowable range of values.

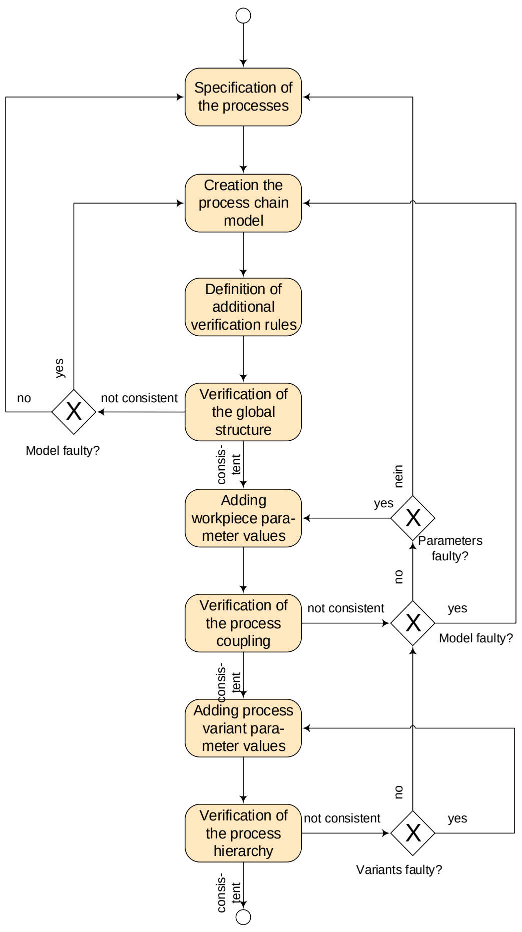

Specification of the processes
Creation the process chain model
Definition of additional verification rules
Verification of the global structure
not consistent
X
no
yes
Model faulty?
consis-tent
Adding workpiece para-meter values
yes
nein
X
Parameters faulty?
no
Verification of the process coupling
not consistent
X
yes
Model faulty?
consis-tent
Adding process variant para-meter values
no
Verification of the process hierarchy
not consistent
X
yes
Variants faulty?
consis-tent

Fig. 3. Modeling and verification steps required to create a consistent process chain model.

The consistency check starts with the verification of the graph structure according to the rules described in Subsect. III.A. Technically, the verification can be realized by two depth first searches [9] over the graph G that represents the process chain. One depth first search is performed in the direction of the graph edges starting with the start node and one depth first search is performed against the edge direction starting with the stop node.

If the graph structure is not consistent because of a faulty model, the process chain model has to be revised. Errors in process chain models occur, for example, if processes are not integrated or dependencies are missing or incorrect. If no errors are found in this revision, the cause of the failed consistency check has to be an incomplete or faulty process specification.

With a consistent graph structure the properties of the workpieces, i.e., the concrete values of the process parameters are determined and a review of the process coupling is performed, as described in Subsect. IV.B. The process coupling check examines the edges of the graph on which the output values of the respective predecessor process and the input values of the respective succession process are annotated. If the process coupling is non-compliant, it is checked whether the model or the detected values are the reason for this. A major challenge is to ensure the completeness of the values acquired, since the extent and the kind of the required values may not always be known initially. Thus, the revision of the process coupling may require multiple iterations consisting of the determination of the values and the corresponding consistency check.

With consistent process couplings the process variants are added and the process hierarchy is designed, as described in section II.D. If the consistency check with respect to the process hierarchy fails, the process variants and possibly the process model has to be checked. If the entire verification is completed successfully (with the last step being the consistency check for the process hierarchy), the entire process chain model is considered consistent.

V. Related Work

IT system support for the virtual product development mainly comes from product data management (PDM) systems [10]. There exist many different PDM systems where most of them are commercial and tailored to the needs of individual enterprises. As a consequence, currently there does not exist a widely accepted model for production process chains. A data model for the representation of the properties and parameters of single processes has been developed in [11]. This model is suitable for the integration into PDM systems, but does not capture the global dependence structure between the processes.

Related work also comes from the modeling of business processes and workflows. Business processes describe the interaction of the various tasks performed by an enterprise or public administration with the goal to produce a certain product or to provide a public service. The corresponding business process models usually focus on the persons and departments responsible for the tasks, the cooperation between different departments and on the documents associated with the tasks. An example for such a model are event-driven process chains (EPC) [3] that consist of an alternating sequence of tasks and events where the events trigger the execution of the tasks, which in turn generate new events . From the technical point of view, an EPC is an annotated directed graph with different types of nodes. The consistency of an EPC can be verified using a set of rules that first require the transformation of a given EPC into an equivalent Petri net [12]. An alternative to EPCs is the graphic modeling language BPMN (Business Process Model and Notation) [4], which supports a large variety of different objects and interaction patterns between these objects. For example, these BPMN objects can represent processes and the interactions can be dependence relations between these processes. In contrast to EPCs and BPMN, the production process chain model proposed in this article is tailored to model production processes and the specific properties of these processes. Workflow languages such as XPDL [5] and WS-BPEL [6] focus primarily on the automated or semi-automated execution by a workflow management system. In contrast, our model focuses on the planning and optimization of production processes, i.e., the representation of different variants for processes with their associated properties.

A mathematical optimization model with the goal to minimize the energy usage of a production process chain has been defined in [13]. A production process chain is modeled as a directed acyclic graph in which different process variants correspond to different paths in the graph. The underlying model does not capture workpiece properties, adjustable process parameters and resources apart from the energy as it is supported by the model proposed in this article.

VI. Conclusion

In this article, we have defined an abstract model for production process chains and have illustrated the model using an example fragment from a real-world process chain. The process chain model is based on a hierarchical directed acyclic graph whose nodes and edges are annotated with the properties of the processes and of the workpieces used. The annotations are flexible in order to enable the definition of a large variety of different processes and to ensure the applicability for future production processes. The model also supports the definition of multiple variants for a production process that may differ in the energy or resource usage. Thus, the model can be used as a basis for optimizing the production, e.g., the selection of process variants that lead to a minimum overall energy use. We have also shown how to check the consistency of a given process chain model using a set of verification rules. These rules can be checked automatically and are therefore an important basis for the integration of the model in an IT system.

Acknowledgment

This work was supported by the European Union (European Regional Development Fund) and the Free State of Saxony in the Cluster of Excellence “Energy-Efficient Product and Process Innovation in Production Engineering” (eniPROD®).

References

[1] J. Niemann, S. Tichkiewitch, and E. Westkämpfer, *Design of Sustainable Product Life Cycles*, Berlin, Heidelberg: Springer Verlag, 2009.

[2] H. J. Jacobs and H. Dürr, “Entwurf und Gestaltung von Fertigungsprozessen. Planung und Steuerung der spanenden Teilefertigung,” *Fachbuchverlag Leipzig*, 2002.

[3] G. Keller, M. Nüttgens, and A. W. Scheer, “Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK),” *Veröffentlichungen des Instituts für Wirtschaftsinformatik Saarbrüclen*, vol. 89, 1992.

[4] Business Model and Notation (BPMN), Version 2.0.1, Object Management Group (OMG). (2013). [Online]. Available: http://www.omg.org/spec/BPMN/2.0.1/

[5] XML Process Definition Language (XPDL), Version 2.2, Workflow Management Coalition (WfMC). (2012). [Online]. Available: http://www.xpdl.org/standards/xpdl-2.2/XPDL%202.2%20(2012-08-30).pdf

[6] Web Services Business Process Execution Language *(WS*-BPEL), Version 2.0, Organization for the Advancement of Structured Information Standards (OASIS). (2007). [Online]. Available: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf

[7] A. Schubert, S. Goller, D. Sonntag, and A. Nestler, “Implementation of energy-related aspects into model-based design of processes and process chains,” in *Proc. 2011 IEEE International Symposium on Assembly and Manufacturing (ISAM),* 2011.

[8] *Industrial Automation Systems and Integration - Product Data Representation and Exchange - Overview and Fundamental Principles*, International Organization for Standardization ISO TC184/SC4, 2004.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2009.

[10] M. Eigner and R. Stelzer, *Product Lifecycle Management - Ein leitfaden für Product Development und Life Cycle Management*, Berlin Heidelberg: Springer Verlag, 2009.

[11] G. Rünger, A. Schubert, S. Goller, B. Krellner, and D. Steger, “Integrating energy-saving process chains and product data models,” in *Proc. Glocalized Solutions for Sustainability in Manufacturing, the 18th CIRP Conference on Life Cycle Engineering (LCE 2011),* Springer, 2011, pp. 519-524.

[12] W. V. D. Aaalst, “Formalization and verification of event-driven process chains,” *Information and Software Technology*, vol. 41, no. 10, pp. 639-650, 1999.

[13] A. Fischer, C. Helmberg, and G. Reghenspurgher, “Energy-Sensitive process chain optimization on the example of forging,” in *Energetisch-Wirtschaftliche Bilanzierung Und Bewertung Technischer Systeme*, R. Neugebauer, U. Götze, and W. G. Drossel, Eds. Wissenschaftliche Scripten, Auerbach, 2013, pp. 311-325.

Jörg Dümmler received a doctoral degree in computer science from the Chemnitz University of Technology in 2010. Since then, he has been working as a postdoctoral researcher at this institution. His current research interests include high-level parallel programming models, mixed and hybrid parallel algorithms, and scheduling for parallel applications.

Sven Gehre received a diploma in mathematics with minor computer science from the Chemnitz University of Technology in 2012. Since then, he has been working as a research assistant at this institution. His current research interests include modeling of energy-sensitive process chains and their verification on the basis of annotated acyclic graphs.

Gudula Rünger received a doctoral degree in mathematics from the University of Cologne in 1989 and a Habilitation in computer science from the Saarland University, Saarbrücken, in 1996. From 1997 to 2000 she has been a professor of parallel computing and complex systems at the University Leipzig. Since 2000 she is a full professor of computer science at Chemnitz University of Technology. Her research interests include parallel and distributed programming, parallel algorithms for scientific application as well as software development in science and engineering.