Hierarchical modelling and verification based on Petri net components with multiple import interfaces

U. Küssel * **J. Padberg** ** **D. Abel** ***

* *Institute of Automatic Control, Technical University of Aachen, 52074 Aachen, Germany (Tel: 0241-8027479, e-mail: u.kuessel@irt.rwth-aachen.de)*
** *Institut für Softwaretechnik und Theoretische Informatik, Technische Universität Berlin, 10587 Berlin, Germany (Tel: 030-31424165, e-mail:padberg@cs.tu-berlin.de)*
*** *Institute of Automatic Control, Technical University of Aachen, 52074 Aachen, Germany*

Abstract: This paper introduces the modelling of discrete event based system and the verification of their properties using Petri net components. Especially interesting is the application of a component based verification approach in order to structure the Petri nets hierarchically and to verify their properties component-wise. Here, we exemplify a new notion of the theory that facilitates modelling. This extension allows the definition of multiple import interfaces. Multiple import interfaces allow the component to import more than one other component and so simplifies the task of the modeler as it provides means for a "divide and conquer" strategy.

Keywords: Discrete event systems modeling and control, Petri Nets and other tools.

1. INTRODUCTION

Process modelling and analysis is a crucial step to controller design of technically controlled systems. A key role of the modelling procedure is to understand the process in more detail as well as to determine the needed accuracy of modelling. Nevertheless, a model is also established for the purpose of analysis and verification of system properties. In the field of discrete event based systems modelling using Petri nets is common practice due to their capability of dealing with concurrency and due to their well established analysis methods, see e.g. Abel and Bollig (2006). Large scale processes are still difficult to model without any structuring methods; it is especially error prone and uncomfortably for the modeler. Varied concepts of hierarchical modelling with Petri nets exist, but usually the possibilities of analysis decrease or the structuring has only visual character and analysis has to be done for the unfolded net. The work by Kindler and his group on component tools Kindler et al. (2006) supporting the definition of components with different underlying formal models in different notations at different levels of abstraction is given at the tool level and less directed to modelling of discrete event based systems. For the field of controls of discrete event based systems we suggest a component based hierarchical modelling, applying Place/Transitions nets without the loss of important analysis and verification methods. The concept of component based Petri nets relies on the component concepts of Continuous Software Engineering Weber (1999) Große-Rhode et al. (2000). According to the concept the body of a component is extended by two interfaces, the import and the export. Hereby, the body net BOD describes the internal desired functionality of modelled subsystems, the import IMP states prerequisites of into the body net integrated components and the export displays the behaviour of the body net in an abstract form. The import-export implication of the Petri net components expresses properties of the abstract export net that are guaranteed provided the imported environment satisfies the import requirements. The import-export implication is expressed by a temporal logic formula. Therefore a suitable temporal logic calculus is need as for example in Girault and Valk (2003). The underlying idea is that components guarantee specific properties for the export net if import assumptions are satisfied Padberg (2006). The hierarchical composition of components requires that the corresponding interfaces, namely EXP and IMP, coincide. Then, as shown in Padberg et al. (2007) and illustrated in Padberg and Küssel (2007), we have compositional verification in the sense that the import-export implications can be composed according the composition of the components. In order to simplify this approach for the modeler we develop multiple interfaces that even may overlap. Multiple import interfaces are very useful as they allow using different components as the environment. Hence, the hierarchy is not merely a sequential but a tree-like structure. In order to keep the advantages of the compositional verification as given by the hierarchical composition we introduce the partial composition of components. This composition allows using only one of several import interfaces. This approach is investigated by considering a Petri net based sequence

controller developed for a model plant in the field of manufacturing engineering. The controller is modelled using the tool Netlab as introduced in Orth et al. (2006). This tool is used for modelling, analysis and simulation of discrete event based systems applying P/T Petri nets.

The contribution is structured in the following way. In Section 2 the benchmark process, a model plant of a manufacturing process, is presented. Section 3 introduces the concept of component based hierarchical modelling and verification with a focus on multiple and even overlapping interfaces. There the Petri net model of a subsystem of the controlled process is given as an example of the introduced notions. Beginning with the flat net, components are extracted in order to structure the net hierarchically. In addition multiple import interfaces and partial composition are motivated using this example of a technical application.The contribution is closed with section 4 which summarises contents, draws conclusions and shows future work.

2. MODEL PLANT AS BENCHMARK PROCESS AND DERIVATION OF COMPONENTS BASED ON PETRI NET COMPONENTS

The approach of a component-based modelling and verification is applied to a technical system to evaluate the possible benefits. As benchmark process a model plant is used for pointing out these advantages especially for complex large-scale processes.

2.1 Manufacturing model plant

The manufacturing model plant is situated at the Institute of Automatic Control (IRT) at the Technical University of Aachen (RWTH Aachen). The discrete event based process of the model plant describes the packing procedure of a liquid product coming from an arbitrary process plant. The model plant can be divided into three partitions as depicted in Fig. 1 which shows the overview of the plant.

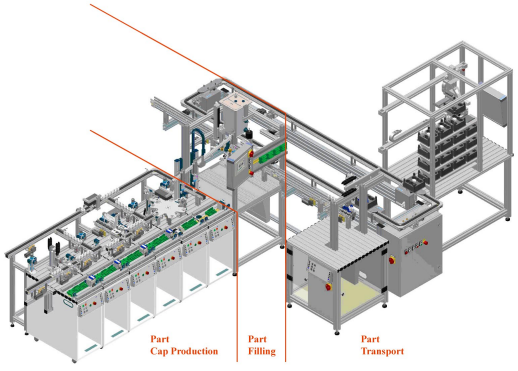

Fig. 1. Model plant for packing process

The left partition produces the closure heads for the glass bottles, in the following denoted as caps ("Cap Production"). The middle partition is responsible for the filling of the liquid product of the process plant from storage into provided glass bottles. Thereafter, the bottles are closed with the supplied caps from the left partition of the model plant. The middle partition is named "Filling". The right most partition ("Transport") is grouping six filled bottles into trays, the so called six-packs. These six-packs are transported on a circular band conveyor which connects the six-pack packing station with the high rack storage area. Both parts are located on the very right side of the model plant in the partition "Transport". Although the model plant can be characterised by these three partitions, every partition by itself is structured in so called stations. The right partition "Transport" is divided into three stations, namely the "Packing Station" (six-pack packing), the "Transport Station" (transport on the circular band conveyor) and the "High Rack Storage Station". The middle partition "Filling" is one station by itself, called "Filling Station". The left partition "Cap Production" is divided into five stations which will be explored in more detail in the following section. All stations are assembled with a new generation of standard PLC hardware. The stations are ordered linearly to avoid concurrency and resulting complex coding of sequence controller programs. Every station gets status information from the following station by transferring one status bit between the PLC's. The level of complexity is therefore reduced by hardware structuring.

2.2 Partition "Cap Production"

A closer look to the partition "Cap Production" is given in Fig. 2. To the right most the "Pick and Place Station" is handling the produced caps. In case that a quality criterion is not met or the "Filling Station" is not activated, the caps will be placed on the feedback band conveyor for either being sorted out or being mixed with new incomplete caps. In case of good quality and activity of the "Filling Station", the caps are handed over. To the left, the "Compression Station" is located. It consists of a turntable which transports the caps to different internal positions where four different tasks are carried out. First the loosely placed RFID chip is pressed into the cap, then the quality of pressing is measured, next the gathered quality information is written onto the chip and finally the chip is released off the table.

The caps are entering the station with a RFID chip (Radio frequency identification data) placed loosely on the top. In a first step the chip is pressed into the cap by a fluidic muscle, followed by a measuring device that detects the quality of the pressing action (quality criterion). Thereafter the information is written on the data chip in order to provide this information for following process steps. Again, to the left locates the "Chip Handling Station" which sites a chip onto a cap in the case that no chip is in position so far. Ahead to this station, the "Lock In Station" is mixing caps from the feedback loop with new incomplete caps from the left most station. In addition, this station is also separating the caps on the feedback loop, depending on the information data written on the RFID chip. The station to the left is called "Turn over". Depending on the orientation of the cap coming from the storage a robot unit picks the cap and turns it over.

The fourth station "Compression Station" is now the object of a more detailed observation as it is used as example for component derivation in Section 3.4.

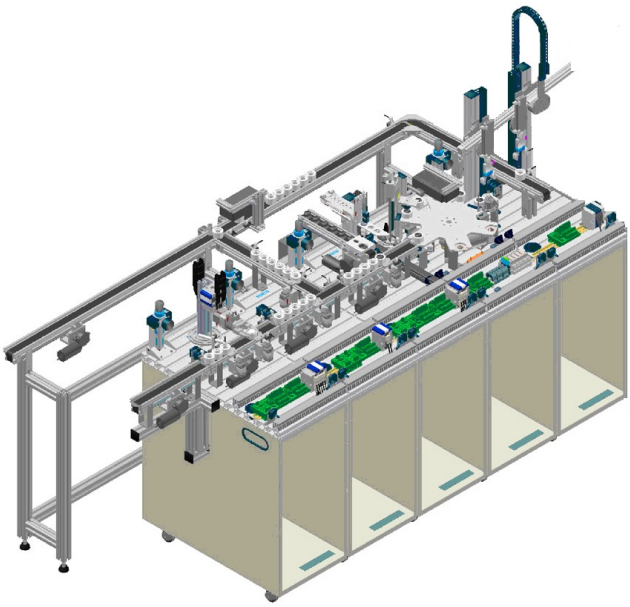

Fig. 2. Station 1 to 5

3. COMPONENT-BASED HIERARCHICAL MODELLING AND VERIFICATION

3.1 Concept of the component based approach

Basically, a component consists of an import interface, an export interface and a body specification. Composition of components is achieved by a hierarchical operation that involves the import interface of the requiring component and the export of the providing component. Accordingly a Petri net component COMP = (IMP,EXP,BOD) consists of three place/transition nets: the import P/T net IMP, the export P/T net EXP and the body P/T net BOD.

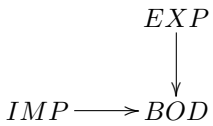

Fig. 3. Single body component with import and export

The interfaces are mapped to the body using suitable mappings of P/T nets as illustrated by the diagram in Fig. 3. In Section 3.4 these mappings are given in terms of the colourings of the places, where the import interface (e.g. in Fig. 8) is illustrated by the red places and the transitions in between. The export interface (e.g. in Fig. 6) is adumbrated by the green places, but conforms to the corresponding import interface, in this case it corresponds to the import interface in the dashed box of Fig. 7.

3.2 Component-Based Verification

Components are self-contained units with a well-defined syntax and semantics. In Ehrig et al. (2002) semantics of components is defined by considering each possible environment expressed by each possible transformation of the component's import. According to the transformation-based semantics the notion of import-export implications characterise the Petri nets component with respect to its environment. Based on a suitable temporal logic calculus that allows the formulation of formulas and import-export implications can be defined. In Padberg et al. (2007) we have extended the component concept with import-export implications of components that are formulas given in that temporal logic. The export statement given as part of the export interface is guaranteed independently of the component's environment provided the import requirement is met. This approach to component verification helps to guarantee specific properties that are formalised in terms of a temporal logic. The underlying idea is that components guarantee specific export statements provided that the import assumptions are satisfied. So, components are equipped with an additional import-export implication $\rho \Longrightarrow \gamma$ where ρ is a temporal logic formula concerning the component's import and γ is a temporal logic formula over the component's export. The component guarantees γ provided that ρ holds. The satisfaction of the import-export implication by a component is formulated with respect to an arbitrary environment. For the hierarchical composition of a requiring component and a providing component the export statement of the providing component has to imply the require assumptions of the requiring component's import. So, in case of hierarchical composition of two components the import-export implications can be combined if the providing component meets the import requirements of the requiring component (for details see Padberg et al. (2007)). Then the result of the composition is a component that guarantees the original exports statements of the requiring component if the import assumptions of the providing component are met.

3.3 Multiple Interfaces

The new concept we illustrate here are multiple import interfaces. These imports may have overlapping places and hence influence each other. Nevertheless, we can extend the concept of import-export implications and so have means for the verification of Petri net components as introduced in Padberg et al. (2007) in case of multiple import interfaces. Multiple imports IMP_1 to IMP_n that overlap at some places O are given by a gluing construction that glues the import interfaces along the overlapping places and express the split import IMP as shown in the diagram in Fig. 4. The formal description as given in Padberg (2007) uses different kinds of mappings and a few categorical constructions, mostly pushouts.

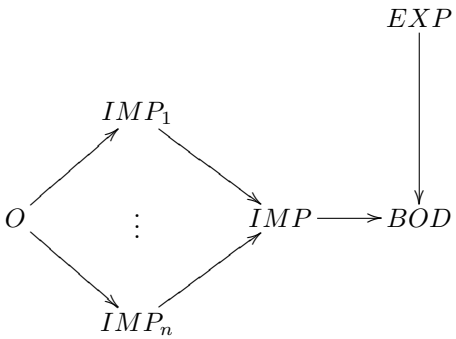

Fig. 4. Component with multiple imports

The partial composition allows the splitting of the import into two (or more) import parts IMP_1 to IMP_n, provided

the imported component's import can be glued adequately to those import parts that are not used. The case of non-overlapping import interfaces is merely a special case of the split import. The partial composition allows connecting one of the import interfaces, e.g. import IMP_1 to an export interface of another component and results in a new component that again has a split import. This new split import consists of the import parts of the first component that have not been used IMP_2 to IMP_n and the import parts of the imported component. According to Ehrig and Mahr (1990) the ordering of the partial compositions along different import parts is irrelevant. The main result for the successful application of this new composition operation for components is that it allows component-based verification as well. Given several import interfaces IMP_n then we have an import assumption that consist of a conjunction of import requirements $\rho = \rho_1 \wedge \rho_2 \wedge \ldots \wedge \rho_n$ so that for each import there is exactly one import requirement ρ_i. To achieve component based verification we have shown in Padberg (2007) that the partial composition of Petri net components again yields a component with guarantees, i.e. a new component that again satisfies its import-export implication. We illustrate a Petri net component using colourings of the net elements, e.g. the net component in Section 3.5.

3.4 Hierarchical, Petri net based modelling of station "Compression"

The Petri nets graphically illustrated in this section show the body specification. The interfaces corresponding to the body are pointed out by using colourings of the net elements. Thereby the export is denoted by green nodes, the import is given by red nodes and overlapping interface nodes, i.e. places, are stressed explicitly by highlighting circles. The original Petri net model of Station "Compression" is depicted in Fig. 5.

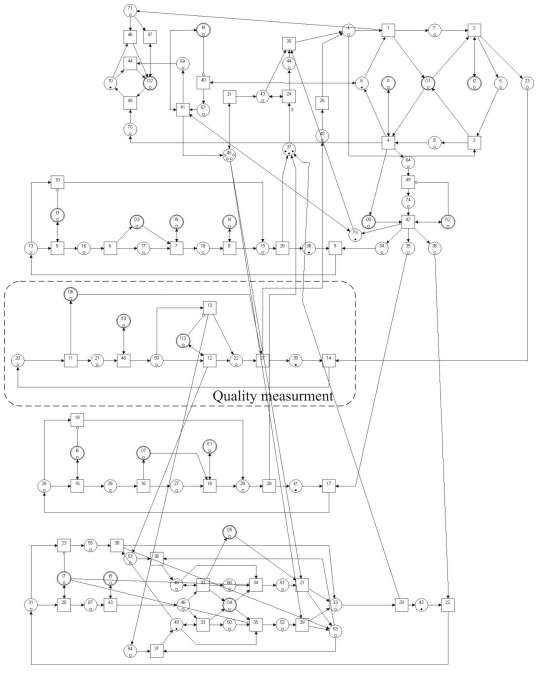

Fig. 5. Original Petri net of Station "Compression"

It is easy to see that the graphical representation suffers from the net size although only a small part of the plant is modelled. The controlled process is now fragmented into 3 hierarchical layers (L0, L1, L2). The bottom layer L2 consists of different nets which describe various operations at the internal positions of station "Compression". There exist four different tasks, namely pressing the loosely placed RFID chip into the cap, measuring the quality of pressing, writing the gathered quality information onto the chip and last but not least releasing the chip off the table. As an example, the net for quality measurement is depicted in Fig. 6.

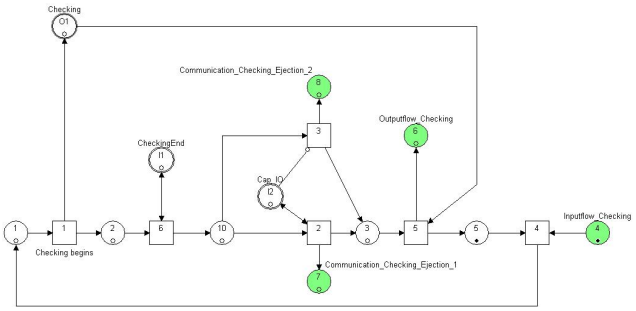

Fig. 6. Layer 2 body net of quality measurement

At the lowest layer there are no import interfaces and as a result no red nodes are given. In addition, the green nodes indicate that this net is incorporated into a higher layer in hierarchy. The corresponding export of this net can be found inside the dashed box in Fig. 7 where it is marked as import.

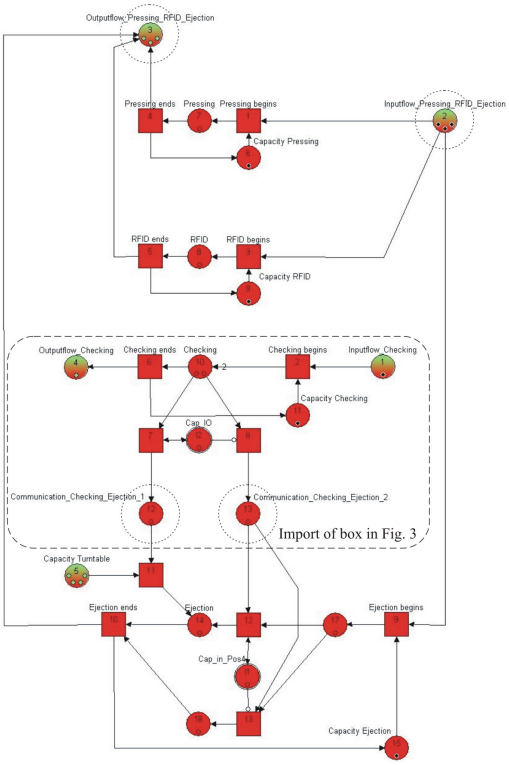

Fig. 7. Layer 1 body net of abstract actions

This net summarises all four functionalities in a more abstract way. It is important to notice that the import interfaces (red nodes) of this component are identical to

all the export interfaces at the lower level components. The small dotted circles point to nodes which represent overlapping interfaces. Overlapping interfaces mean that places of different export interface are glued together in places of the import interface. An overall import interface with an independent or parallel set of import components is possible, although it is not shown here. In addition, there are places which are part of the import and export at the same time. This is stressed with a fading in colour from green to red. The information of these places is simply handed over from top layer to bottom layer. In Fig. 8 the up most net of the hierarchical approach is displayed.

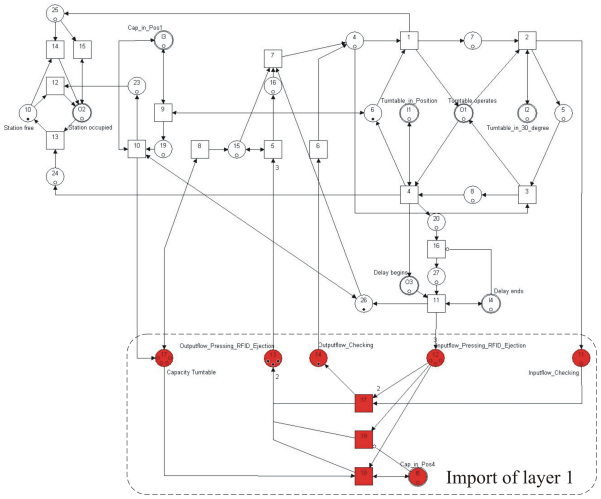

Fig. 8. Layer 0 body net of station "Compression"

Since this is the upper end of hierarchy, there are no nodes of export interfaces to display. Nevertheless, the component of the level below is glued into the net using its export representation. Again the red nodes denote the import interface which must equal the export representation of the corresponding net in Fig. 7.

3.5 Component-based Verification

So far the concept of a hierarchical modelling using Petri nets is applied to a technical system. Another aspect needs to be pointed out. The concept of component based hierarchical modelling also provides import/export implications in order to verify desired properties throughout the layers even if multiple interfaces with eventually overlapping places are used. In terms of this technical benchmark the modeler is interested in verifying certain properties of the controlled system. Using a bottom up strategy one needs to guaranty that for a given initial marking of input and body places a desired final marking for output and body places is reached. If the overall process is cyclic and the lower layer is called several times, the initial marking of body places needs to be preserved. As a result a desired deadlock concerning the output places is to be guarantied while a pseudo- reversibility of the body places is desired. These properties can be easily expressed in some temporal logic calculus. The desired quasi-reversibility is more or less the same as soundness in workflow nets van der Aalst (1997). One layer up, one requires this firing behaviour for a given input marking of the imports. The same properties are now to be verified for the export. Finally, in the up most layer real deadlock avoidance and reversibility of the cyclic system can be verified. In other words the concept of import/export implications makes possible to guarantee the export property formulated as γ as long as the body and the imports fulfil the required property ρ.

4. CONCLUSIONS

In this contribution the concept of component based modelling and verification was introduced. A component consists of three nets, namely the body BOD, the import IMP and the export EXP. Using the import and export interfaces a component can be constructed hierarchically by gluing the corresponding interfaces. Desired properties of the flat net can be guarantied for components that satisfy specific import/export implications. An essential result is that the concept still holds if multiple imports are used, even for overlapping places. A part of the benchmark process plant was presented and modeled by one Petri net for the Station "Compression". Using this net, a hierarchical structure with 3 layers was constructed. A bottom up strategy was used for developping the body nets from the original net. In a next step the export representation of these nets was derived. Only the body nets were depicted completely while the import and export interface nodes were denoted using colours, red and green respectively. It is easy to notify that the hierarchical structure divides the complex flat net into smaller, more manageable nets whose properties are more efficient and with less effort to be verified. Especially, the use of import/export implications for the need of verification reduces the system to subsets which can be analysed and then lead to general results of the complete system.

This leads to more efficient verification procedure due to smaller reachability space in subsystems. This example therefore shows that the modeler can benefit of such a strategy for modelling and verifying large scale systems with Petri nets while using methods of computer science in the field of process engineering. Nevertheless, there are some obstacles to overcome in order to apply this method in a powerful manner in the field of process engineering. As described in this contribution, an existing net was divided in hierarchical layers in order to achieve the named benefits. The idea is to motivate a new approach for modelling and verification applying a top down or bottom up strategy. In the case of bottom up modelling, the modeler could be supported by an algorithmic procedure for the design of the abstract export representation of the body net since this reduction of the system is a crucial step in the design process. This algorithm procedure would also be useful for verifying nets designed in a top down approach. For applying this concept technically it still has to be implemented.

REFERENCES

D. Abel and A. Bollig, editors. *Rapid Control Prototyping.* Springer, Berlin, 2006.

H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 2: Module Specifications and Constraints*, volume 21 of *EATCS Monographs on Theoretical Computer Science.* Springer, Berlin, 1990.

H. Ehrig, F. Orejas, B. Braatz, M. Klein, and M. Piirainen. A Generic Component Concept for System Modeling.

In *Proc. FASE 2002: Formal Aspects of Software Engineering*, *Lecture Notes in Computer Science 2306*, pages 32–48, 2002.

C. Girault and R. Valk, editors. *Systems Engineering: a Guide to Modelling, Verification and Applications*. Springer, 2003.

M. Große-Rhode, R.-D. Kutsche, and F. Bübl. Concepts for the evolution of component-based software systems. Technical Report 2000/11, TU Berlin, 2000.

E. Kindler, V. Rubin, and R. Wagner. Component tools: Integrating Petri nets with other formal methods. *Lecture Notes in Computer Science 4024*, pages 37–56. Springer, 2006.

Ph. Orth, U. Küssel, and D. Abel. Netlab und Netlab Toolbox für Matlab/Simulink - Ein Werkzeug zum Rapid Control Prototyping von Steuerungen mittels Petrinetzen. In *Tagungsband Entwurf komplexer Automatisierungssysteme EKA 2006*, pages 343–353. Institut für Regelungs- und Automatisierungstechnik TU Braunschweig, 2006.

J. Padberg. Regelbasierte Verfeinerung von Petri-Netz Modulen. In *Tagungsband Entwurf komplexer Automatisierungssysteme EKA 2006*, pages 57–67. Institut für Regelungs- und Automatisierungstechnik TU Braunschweig, 2006.

J. Padberg. Partial composition of components: Formal foundation and component verification. Technical Report, Technische Universität Berlin,Fakultät IV, 2007.

J. Padberg and U. Küssel. A component-based verification approach based on Petri net components. In *Proc. FORMS/FORMAT 2007 - "Formal Methods for Automation and Safety in Railway and Automotive Systems"*, pages 40–50. GZBV, 2007.

J. Padberg, H. Ehrig, and F. Orejas. Towards component verification in the generic component framework. In J. Kuester-Filipe, I. Poernorno, and R. Reussner, editors, *Proc. Formal Foundations of Embedded Software and Component-Based Software Architectures (FESCA 07)*, Electronic Notes in Theoretical Computer Science, Amsterdam, 2007. Elsevier Science. to appear.

W.M.P. van der Aalst. Verification of workflow nets. *Lecture Notes in Computer Science 1248*, pages 407 – 426, 1997.

H. Weber. Continuous engineering of information and communication infrastructures. *Lecture Notes in Computer Science 1577*, pages 22–29. Springer, 1999.