Functional Analysis of Process-Oriented Systems*

Peter Buchholz and Carsten Tepper

Dept. of Computer Science IV, University of Dortmund, D-44221 Dortmund
e-mail:{peter.buchholz,carsten.tepper}@udo.edu

Abstract. A major problem in modelling and subsequent simulation of process-oriented systems (ProC/B models), is the functional correctness of the model. Therefore a model should be first analysed for its functional correctness before it is analysed by simulation. Petri nets are well suited for model based and state based functional analysis, but are often not adequate or not used for the specification of process models. We present in this paper a transformer for an automatic mapping from ProC/B models onto PNs. The resulting PN-models can be analysed with PN-algorithms and the results from the PN-analysis can be interpreted at the ProC/B level.

1 Introduction

To find optimal or at least good designs and strategies for large logistic networks often a model based approach is necessary. Usually the networks are modelled using some process oriented approach like process chains [5]. Ideally the model should be usable for the automatic analysis of performance, reliability and correctness of the logistic network. One formalism which allows an automatic mapping of the process chain model onto a simulator is the ProC/B formalism [1] which has been developed together with a corresponding toolset within the Collaborative Research Center 559 'Modelling of Large Logistics Networks'.

One major problem of modelling and subsequent simulation is the functional correctness of the model. Simulation can only hardly detect functional errors like deadlocks or livelocks in a system such that the simulative analysis of an incorrect model can result in strange and incorrect simulation results. Therefore a model should be first analysed for its functional correctness before the simulation is started. Unfortunately, process oriented descriptions like ProC/B often do not support functional analysis. On the other hand Petri nets are well suited for functional analysis of discrete event systems and a large number of analysis algorithms are available. Thus, a natural solution is to map the process chain model onto a PN. In this paper, we present an automatic mapping of ProC/B models onto Petri nets (PNs) show how different functional analysis algorithms can be applied to this model and indicate which results for the process chain can be derived from the PN and where are the limits of the approach.

The paper is structured as follows. In the next section we give a brief overview of the ProC/B formalism. Before introducing the translation of

* This research is supported by DFG, collaborative research centre 559 'Modelling of Large Logistic Networks'

ProC/B models onto PNs in section 4, some basic definitions for PNs are briefly introduced in section 3. Section 5 explains the functional analysis of ProC/B models via PN analysis. In section 6 we show an example of a store for a freight village and conclude the paper with a summary and an outline of future work.

2 ProC/B formalism

A modelling language which is especially designed to the needs of logistic networks is the ProC/B formalism, which is accompanied by a corresponding software toolset including a graphical interface for specification and a set of analysis tools [1]. Table 1 includes the core ProC/B modelling elements. The ProC/B formalism has proved its value in several industrial projects and it is the joint specification language in a large, interdisciplinary, and longterm collaborative research centre (SFB 559) at the university of Dortmund [1].

The main structuring elements of ProC/B-models are *Functional Units* (FUs), which encapsulate one or more *Chains.* A chain can be viewed as a structured and measured set of activities starting with *Sources* for process creations (demand), denoted by a circle with midpoint ⨀, followed by a chronological sequence of *Process Chain Elements* (PCEs) describing activities, denoted by arrow-like hexagons, and the chain is accomplished by a sink, denoted by ⨂. Horizontal connections between PCEs indicate a sequential behavioural pattern of processes. Branches into and merges from alternative sub-chains are allowed and represented by vertical bars. *Process Chains* can be hierarchically structured. PCEs may invoke sub-chains from so-called *subordinated FUs.* Subordinated FUs and sub-chains are described in the same manner as their super-ordinated FUs and chains (self-similarity). Actually, two types of subordinated FUs are distinguished: user-defined FUs and simple, predefined FUs of type *Server* or *Counter.* Servers are used to model active, possibly shared resources, i.e. machines, assembly lines, workers. In principle, servers correspond to single stations in queueing networks. Counters are used to describe passive resources, i.e. stores and waiting areas of usually restricted capacity.

3 Basic Definitions of Petri nets

First, a brief introduction to Petri nets (PNs) and some associated terminology is provided before we describe the translation of ProC/B onto PNs. Only some concepts which are relevant for understanding the rest of the paper are addressed. More complete material can be found in the literature e.g. [6].

A Petri net is a bipartite directed graph composed of places, drawn as circles, and transitions, drawn as rectangles. As usual, a directed graph is formally given by the description of its elements and functions or matrices specifying their interconnection.

Definition 1. A Petri Net (PN) is an 5 tuple $PN = (P, T, I^-, I^+, M_0)$, where $P = \{p_1, \dots, p_n\}$ is a finite and non-empty set of places, $T = \{t_1, \dots, t_m\}$

[1] see `http://www.sfb559.uni-dortmund.de/index.php?lang=eng`