Rule-Based Refinement of High-Level Nets Preserving Safety Properties

J. Padberg, M. Gajewsky, C. Ermel
e-mail: {padberg, gajewsky, lieske}@cs.tu-berlin.de

Technical University of Berlin**

Abstract. The concept of rule-based modification developed in the area of algebraic graph transformations and high-level replacement systems has recently shown to be a powerful concept for vertical stucturing of Petri nets. This includes low-level and high-level Petri nets, especially algebraic high-level nets which can be considered as an integration of algebraic specifications and Petri nets. In a large case study rule-based modification of algebraic high-level nets has been applied successfully for the requirements analysis of a medical information system. The main new result in this paper extends rule-based modification of algebraic high-level nets such that it preserves safety properties formulated in terms of temporal logic. For software development based on rule-based modification of algebraic high-level nets as a vertical development strategy this extension is an important new technique. It is called rule-based refinement. As a running example an important safety property of a medical information system is considered and is shown to be preserved under rule-based refinement.

Keywords: Petri nets, high-level nets, algebraic specification, safety property, rule-based refinement

1 Introduction

Petri nets are well-known as a basic model for the general theory of concurrency and as a formal specification technique for distributed and concurrent systems. High-level nets can be considered as the integration of process and data type description, most prominent classes are Coloured Petri nets [Jen92, Jen95], Predicate/Transition nets [GL81, Gen91] and algebraic high-level nets [Vau87, Rei91, PER95]. The practical relevance of high-level Petri nets is considered to be very high, as there are many high-level Petri net tools used in real software production (e.g. LEU [SM97] , Design/CPN [JCHH91], INCOME [OSS94]). Since algebraic specifications are well developed for abstract data types (see e.g. [EM85]) we use algebraic high-level nets, but there is no problem of transfering results to other

** This work is part of the joint research project "DFG-Forschergruppe PETRINETZ-TECHNOLOGIE" between H. Weber (Coordinator), H. Ehrig (both from the Technical University Berlin) and W. Reisig (Humboldt-Universität zu Berlin), supported by the German Research Council (DFG).

high-level net classes as these classes can be conceived as different instances of a general theory of abstract Petri nets (see [Pad96]).

One main problem of verification in formal software engineering can be described by the following demand: Rigorous software development requires continuous verification during all phases of the software development process. Nevertheless, resources are restricted and an entirely new verification at each step is usually considered to be too expensive and time consuming. Thus, vertical structuring techniques should preserve verified properties.
In the area of Petri nets there are many contributions concerning verification with temporal logic [DDGJ90, BS90, HRH91] and refinement [BGV90, DM90, GG90, BDH92, Peu97]. They are mainly in the area of low-level nets. In the area of high-level nets, verification [Jen95, Sch96] is much more difficult and even more the compatibility of system properties with refinement.

In this paper we consider our notion of rule-based modification of algebraic high-level nets (developed in [PER95]) and extend it to rule-based refinement preserving safety properties. The theory of rule-based modification is an instance of the theory of high-level replacement systems [EHKP91], a generalization of graph transformation [Ehr79] in a categorical way. Rules describe which parts of a net are to be deleted (left side of the rule) and which new parts are to be added (right side of the rule). This transformation of nets yields a resulting net which is well-defined and no unspecified changes have been made. The advantage of this approach is the local description of change.
In order to extend rule-based modification of algebraic high-level nets we introduce morphisms for algebraic high-level nets, that – in contrast to transition preserving morphisms in [PER95] – preserve safety properties, in the sense of [MP92]. These morphisms, called place preserving morphisms, allow transfering specific temporal logic formulas expressing net properties from the source to the target net. This fact is captured by our first main theorem 3.5 that states the fact that place preserving morphisms preserve invariant formulas. As invariant formulas describe safety properties we hereby obtain safety property preserving algebraic high-level net morphisms.
Moreover, we combine these place preserving morphisms with rule-based modification. The second main result of this paper is formulated in theorem 4.2. It states the preservation of safety properties under transformation of nets via some rule that is provided with such a safety property preserving morphism. This allows the formulation of the new concept 4.1 that is the extension of rule-based modification to rule-based refinement, a formal technique for vertical structuring in software development.

Throughout the whole paper we give an ongoing example which illustrates the results of this paper in the context of a case study [Erm96, EPE96] concerning the development of a medical information system. A sketch of this case study as well as a review of the basic notions of algebraic high-level nets and rule-based modification is given in the next section. In section 3 we introduce the notion of place preserving morphisms. Our first main result states that these morphisms preserve safety properties. In section 4, rule-based modification is

integrated with these morphisms. We present our second main theorem, showing that rule-based refinement preserves safety properties. Moreover, we discuss the relevance of our results for software engineering, especially the combination of horizontal structuring and refinement.
In this paper we merely give the proof ideas due to space limitations, in full detail the proofs are given in [PGE97].

2 Rule Based Modification and Safety Properties in a Medical Information System

In this section we sketch our case study and motivate the notions and results of the subsequent sections in terms of this case study. The motivation addresses general problems in software engineering. Any large and complex system can only be developed using horizontal and vertical structuring that is, stepwise development of subsystems. This implies that the entire system is given only implicitly. Thus, verification has to be achieved according to horizontal and vertical structuring. We now show an example of verifying a safety property, first in a net and then for one development step. Note, this is merely a small example from the larger context of the medical information system.

The medical information system HDMS

A medical information system, called Heterogeneous Distributed Information Management System (HDMS), has been developed in a large project, that included the whole reorganisation of the medical and management data of the German Cardiac Center Berlin, Deutsches Herz-Zentrum Berlin (DHZB). This project has been developed by the Projektgruppe Medizin/Informatik at the DHZB and the Technische Universität Berlin[1]. The DHZB is a clinical center dedicated to the treatment of all kinds of cardiac diseases. In our case study [Erm96, EPE96] we provide a formal requirement analysis for an important part of the medical information system HDMS at the DHZB using algebraic high-level nets. The transformation sequence from the actual state to the functional essence comprises about 100 rules and uses in a significant way compatibility results from [PER95] between horizontal structuring and rule-based modification. Here we present one transformation step of the whole rule-based modification. We demonstrate how safety properties are preserved using a special kind of algebraic high-level net morphisms and the new concept of rule-based refinement.

Example 2.1 (Safety Properties for Vital Values Measurement)

An algebraic high-level (AHL) net can be considered as a Petri net inscribed with terms over a specification, in this case the specification VVM-Spec which is merely sketched below due to space limitations. Tokens are elements of a VVM-Spec-algebra.

[1] The case study HDMS, the basis for our work, has been a part of the German BMFT-project KORSO, (KORrekte SOftware), funded by the Minister of Research and Technology (BMFT) between 1991 and 1994 [CHL95].

Let us shortly explain the idea of the net VVM in figure 1. In the DHZB we have the following situation: The patient is located at the ward. His blood pressure is taken, for example, if this has been demanded in the prescription sheet. The measured value is written down into the temperature chart. Other vital values, as medium arterial blood pressure, temperature, pulse, central venous pressure and import/export are also measured, if demanded in the prescription sheet. The temperature chart belongs to the patient record that is kept at the ward. All these activities are represented as transitions in the net VVM in figure 1. Note, that we restrict our example to this small subsystem concerning the measurement of vital values.

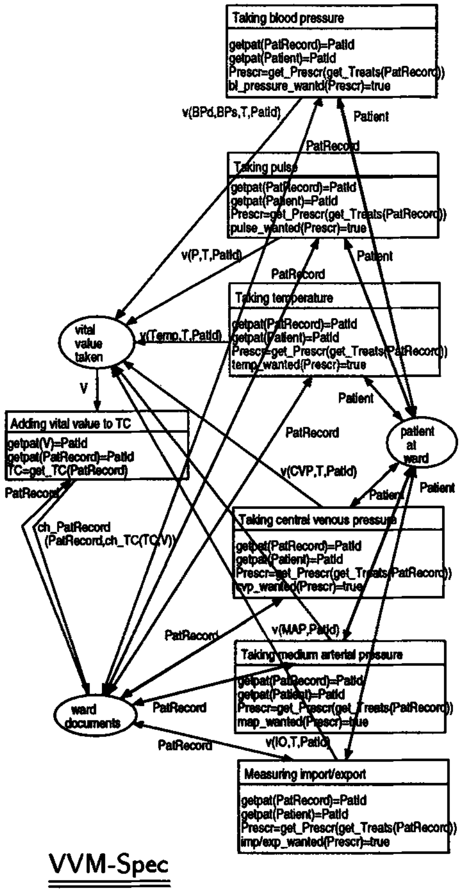

Fig. 1: The Algebraic High-Level Net Vital Values Measurement (VVM)

We merely state the sorts and operations of VVM-Spec used explicitly in the subsequent argument.

sorts: Name, Patient, PatId,
PatRecord, ...
opns: patient: Name, Sex, Adress,
PatId → Patient
getpat: Patient → PatId
getpatient: PatId → Patient
getpat: PatRecord → PatId

In the following, we give the marking and one safety property of the net VVM explicitly. We consider the A-quotient algebra (see [EM85]), that is the algebra generated according to the specification over carrier sets for names, doctors, resources etc. Assuming a carrier set $A_{Name} = \{Smith, Miller, ...\}$ we can suppose the following marking: $(patient(Smith, ...),$ **patient at ward**$)$ $\oplus(d,$ **ward documents**$)$ where $d \in A_{PatRecord}$ with $getpat(d) = getpat(patient(Smith, ...))$.
This marking means that there is a patient *Smith* and his patient record at the ward, represented by tokens $(patient(Smith, ...)$ and d on the places **patient at ward** and **ward documents** respectively.

We consider the safety property $\square[(patient(Smith, ...),$ **patient at ward**$) \Longleftrightarrow (d,$ **ward documents**$)]$ with $getpat(d) = getpat(patient(Smith, ...))$ for some $d \in A_{PatRecord}$ where $\lambda(a, p)$ for $(a, p) \in A \times P$ is an atomic formula (see def. 3.3) and $\square$ the always operator from temporal logic [MP92].

We informally argue that this safety property holds. For each transition except ***Adding vital value to TC*** the patient record is only read, denoted by double arrows with the inscription of a variable of sort *PatRecord*. The transition ***Adding vital value to TC*** changes the record, but by structural induction we can prove that no operation changes the initial patient identity. Thus, after firing of any transition the safety property still holds.

In general we assume a marking of the net VVM

$$M^{\mathsf{VVM}} := \sum_{i=1}^{n}(a_i, \textbf{patient at ward}) \oplus (d_i, \textbf{ward documents})$$

s.t. $getpat(a_i) = getpat(d_i)$ for $a_i \in A_{Patient}$ and $d_i \in A_{PatRecord}$.

The more general formulation of our safety property φ^{VVM} is

$$\Box[(a, \textbf{patient at ward}) \Longleftrightarrow (d, \textbf{ward documents})]$$

s.t. $getpat(a) = getpat(d)$ for $a \in A_{Patient}$ and $d \in A_{PatRecord}$.

This safety property means *"At any time we have: there is some patient at the ward if and only if the corresponding patient record is at the ward."* and holds due to the same argument as above. ◇

Algebraic High-Level Nets and Refinement Techniques

An algebraic high-level net consists – roughly speaking – of a Petri net with inscriptions of an algebraic specification $SPEC$ defining the data type part of the net. The data type in our case study is given by a suitable $SPEC$ algebra. In contrast to other variants of algebraic high-level nets ([DHP91, Hum89, Lil94]) we do not label places with sorts. Note, that the pre and post domain of a transition is given by a multiset of pairs of terms and places. Multisets can be considered as elements of a commutative monoid. Here, we use free commutative monoids for the description of the pre and post domain of a transition as this representation allows a more categorical treatment.

Morphisms between different AHL nets are given componentwise and have to preserve the firing conditions and each transition's pre and post domain. This is characterized by the commutativity of diagrams **(1)** and **(2)** in def. 2.2.

Definition 2.2 (Algebraic High-Level Nets)

- An **algebraic high-level net** $N = (SPEC, P, T, pre, post, cond, A)$ consists of an algebraic specification $SPEC = (S, OP, E)$, a set of places P, a set of transitions T, two functions $pre, post : T \longrightarrow (T_{OP}(X) \times P)^{\oplus}$, assigning to each $t \in T$ an element of the free commutative monoid[2] over the cartesian product of terms $T_{OP}(X)$ with variables in X and the set P of places, a function $cond : T \longrightarrow \mathcal{P}_{fin}(EQNS(SIG))$ assigning to each $t \in T$ a finite set $cond(t)$ of equations over $SIG = (S, OP)$, the signature of $SPEC$ and a $SPEC$ algebra A.
- A **(transition preserving) AHL net morphism** $f = (f_{SPEC}, f_P, f_T, f_A) : N_1 \rightarrow N_2$ is given componentwise by the specification morphism $f_{SPEC} : SPEC_1 \rightarrow SPEC_2$, the functions $f_P : P_1 \rightarrow P_2$ and $f_T : T_1 \rightarrow T_2$ and the isomorphism $f_A : A_1 \xrightarrow{\sim} V_{f_{SPEC}}(A_2)$ on the algebras such that the following diagrams **(1)** and **(2)** commute:

[2] The free commutative monoid implies the following operations on linear sums: $\oplus, \ominus, \leq$.

$$\begin{tikzcd}
\mathcal{P}_{fin}(EQNS(SIG_1)) \arrow[d, "f_E:=\mathcal{P}_{fin}(f^{\#}_{SPEC})"'] \arrow[dr, phantom, "(1)"] & T_1 \arrow[l, "cond_1"'] \arrow[r, shift left, "pre_1"] \arrow[r, shift right, "post_1"'] \arrow[d, "f_T"] \arrow[dr, phantom, "(2)"] & (T_{OP_1}(X_1) \times P_1)^{\oplus} \arrow[d, "(f^{\#}_{SPEC} \times f_P)^{\oplus}:=f_S"] \\
\mathcal{P}_{fin}(EQNS(SIG_2)) & T_2 \arrow[l, "cond_2"'] \arrow[r, shift left, "pre_2"] \arrow[r, shift right, "post_2"'] & (T_{OP_2}(X_2) \times P_2)^{\oplus}
\end{tikzcd}$$

The sets of variables are defined by indexing a fixed set $X_i := (X_{fix_s})_{s \in S_i}$ for $i = 1, 2$. In the following, we will use abbreviations for the mappings of markings (f_M), symbolic markings, that is terms with variables (f_S) and sets of equations (f_E):

$f_M := ((f_{SPEC}, f_A) \times f_P)^{\oplus}, f_S := (f^{\#}_{SPEC} \times f_P)^{\oplus}$ and $f_E := \mathcal{P}_{fin}(f^{\#}_{SPEC})$.

- AHL nets and AHL net morphisms are defining the category **AHL** of algebraic high-level nets.
- The **behaviour** of an AHL net N is given by firing of transitions. Transitions are enabled under a marking $M \in (A \times P)^{\oplus}$ for an assignment $asg : X \to A$ inducing $ASG : (T_{OP}(Var(t)) \times P)^{\oplus} \to (A \times P)^{\oplus}$ with $ASG(term, p) = (\overline{asg}(term), p)$ if $ASG(pre(t)) \leq M$. $Var(t)$ is the set of variables that occur in the firing condition $cond(t)$ and in the pre and post domains $pre(t)$ and $post(t)$ for each $t \in T$. The follower marking M' then is constructed by $M' = M \ominus ASG(pre(t)) \oplus ASG(post(t))$, denoted by: $M[t, asg > M'$. The set of all follower markings is denoted by $[M >$.

△

We review rule-based modification as a vertical structuring technique of Petri nets [PER95]. The idea is to present rules denoting the replacement of one subnet by another without changing the remaining part of the whole net. This has the advantage of a local description of changes inducing global changes without side effects. We consider to have a rule r with a left-hand side net L that is replaced by a right-hand side net R. This rule can be applied to some net N, yielding the new net M. This application of a rule, called transformation, is denoted by $N \overset{r}{\Longrightarrow} M$. The rule is given by $r = (L \leftarrow K \to R)$ where K is a net and $K \to L$ and $K \to R$ are injective AHL net morphisms. Deleted are those parts of the net L that are not in the image of the morphism $K \to L$. Adding works symmetrically, all those parts of R are added, that are not in the image of the morphism $K \to R$. Thus, K denotes the common interface between deleting and adding, that is the part of the rule that has to be present but is not changed by the rule. The transformation $N \overset{r}{\Longrightarrow} M$ is defined using two pushout squares **(1)** and **(2)** in def. 2.3 in the category **AHL**. C is the context net (N after the deletion of items by the rule and before the addition of the new items from R).

Definition 2.3 (Rule and Transformation)
A **rule** $r = (L \leftarrow K \to R)$ consists of two AHL nets L and R (called left and right hand sides of the rule), an AHL net K (called interface) and two injective AHL net morphisms $L \leftarrow K$ and $K \to R$.

$$\begin{tikzcd}
L \arrow[d] \arrow[dr, phantom, "(1)"] & K \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "(2)"] & R \arrow[d] \\
N & C \arrow[l] \arrow[r] & M
\end{tikzcd}$$

A **(direct) transformation** $N \overset{r}{\Longrightarrow} M$ of a net N to M via rule $r = (L \leftarrow K \rightarrow R)$ at the match $L \rightarrow N$ is defined using two pushout squares **(1)** and **(2)** shown in the diagram in the category **AHL**. △

This definition is the technical basis for the vertical structuring technique of rule-based modification. Results concerning parallel and concurrent application of rules and compatibility with horizontal structuring can be found in [PER95].

Example 2.4 (Blood Hypertension Test)
We now want to describe the refinement step that adds an exception in case of blood hypertension. In this case the doctor shall be notified immediately.
The transformation rule $r^{\mathsf{VVM}} : L \leftarrow K \rightarrow R$ in figure 2 describes the refinement of the net VVM depicted in figure 1 by an exception for blood hypertension. For each blood pressure value taken an additional test for hypertension is performed. In case of hypertension the doctor is notified. The transition ***Adding vital value to TC*** is part of the interface K in order to ensure the application of r^{VVM} only in the context of vital value measurement.

The inclusion morphism $K \rightarrow L$ means that the transition ***taking blood pressure*** is deleted. Additionally, the right hand side net R contains the places **values for hypertension test** and **doctor**, and the transitions ***notifying doctor*** and ***taking blood pressure***. The corresponding algebraic specification also has to be adapted coherently and persistently by adding the equations used for the entry of blood hypertension (for details see [PGE97]).

The application of rule r^{VVM} to the net VVM yields the following transformation shown in figure 2: The deletion of the transition ***taking blood pressure*** yields the context net C and the addition of the places **values for hypertension test**, **doctor**, and the transitions ***notifying doctor*** and ***taking blood pressure*** yields the net BEX (short for Blood hypertension EXception).

Now the main problem is the transfer of the safety property φ^{VVM} *"At any time we have: there is some patient at the ward if and only if the corresponding patient record is at the ward."* This transfer should be induced by the rule $r^{\mathsf{VVM}} = (L \leftarrow K \rightarrow R)$. To achieve the rule-based refinement we have to find a property of the rule such that the transformation preserves the safety property. We are looking for proof rules of the following form:

$$\frac{\text{some property for } r^{\mathsf{VVM}},\ \mathsf{VVM}\ \text{satifies}\ \varphi^{\mathsf{VVM}}}{\mathsf{BEX}\ \text{satisfies}\ \varphi^{\mathsf{VVM}}}$$

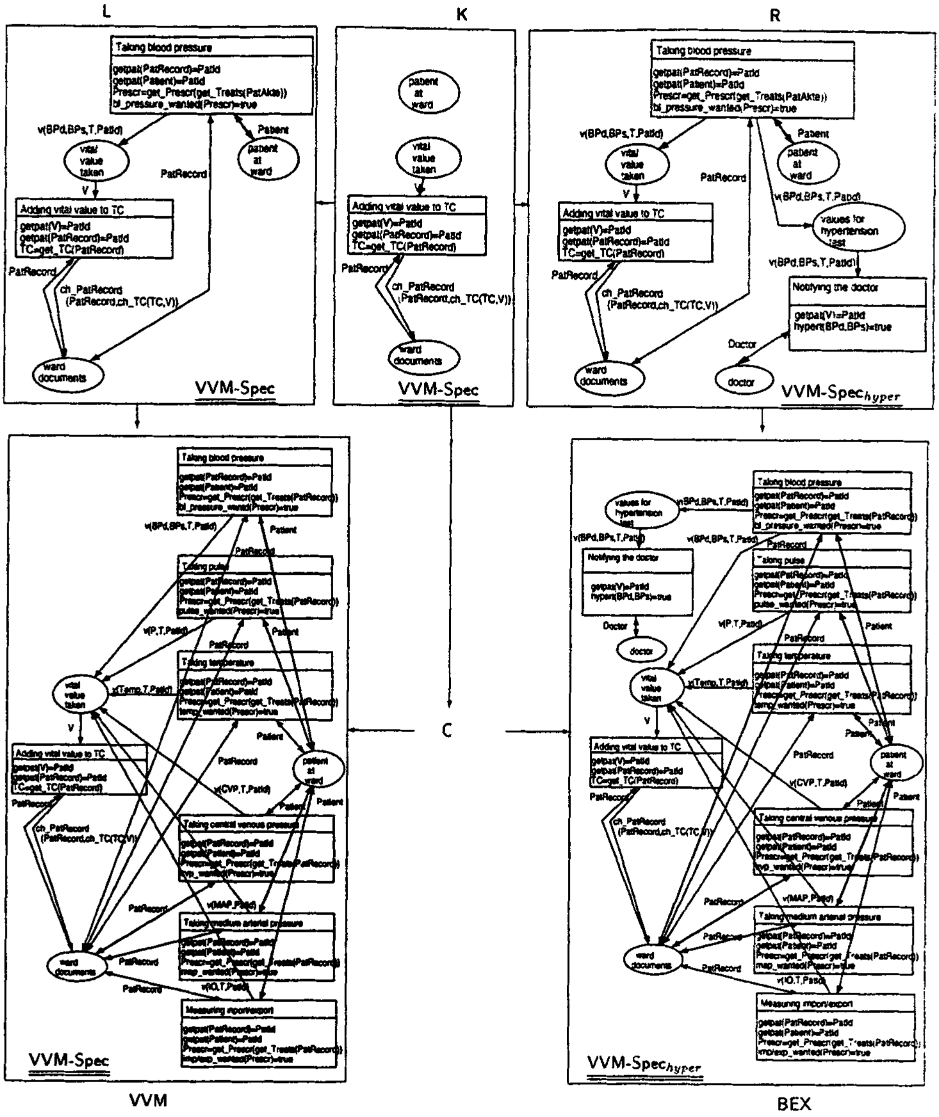

Fig. 2. Vital Value Measurement with Hypertension Exception

$$\begin{tikzcd} L \arrow[d] \arrow[rr, bend left, dotted, "f^{\text{VVM}}"] & K \arrow[l] \arrow[r] \arrow[d] & R \arrow[d] \\ \text{VVM} \arrow[rr, bend right, dotted, "\overline{f^{\text{VVM}}}"'] & C \arrow[l] \arrow[r] & \text{BEX} \end{tikzcd}$$

The main idea of our approach is to use a class of morphisms, called *place preserving*, that on the one hand preserve safety properties (section 3) and on the other hand are stable under transformations (section 4). As a result of a cooperation within the "DFG-Forschergruppe Petrinetztechnologie" (see first page), it has recently been shown in [Peu97] that safety properties are preserved by *place preserving* morphisms for low level nets. In this paper in section 3 we show that the idea can be transferred to high-level nets. Thus, we can transfer safety properties via transformations. The fact that $f^{\text{VVM}} : L \rightarrow R$ preserves safety properties (theorem 3.5) *always* implies that $\overline{f^{\text{VVM}}} : VVM \rightarrow BEX$ preserves safety properties (theorem 4.2). Thus we have the desired property so that the following proof rule holds:

$$\frac{(r^{\text{VVM}}, f^{\text{VVM}} : L \rightarrow R) \text{ preserves safety properties, VVM satisfies } \varphi^{\text{VVM}}}{\text{BEX satisfies } \varphi^{\text{VVM}}}$$

◇

3 Morphisms Preserving Safety Properties

In this section we define morphisms preserving safety properties of algebraic high-level nets. To be able to preserve safety properties (expressed via formulas on markings), we must take care that no new arcs are added to the context of mapped places by the morphism and no old (mapped) arcs are deleted from their context. Otherwise new transitions could add or delete tokens on "old" (mapped) places in an unpredictable way. We therefore call morphisms with these features *place preserving*.

Definition 3.1 (Place Preserving AHL Net Morphism)
Let $N_i = (SPEC_i, P_i, T_i, pre_i, post_i, cond_i, A_i), i \in \{1, 2\}$ be two AHL nets, then $f = (f_{SPEC}, f_P, f_T, f_A) : N_1 \rightarrow N_2$ is called a **place preserving AHL net morphism** if the following holds:

1. Preservation of firing conditions: $f_E \circ cond_1 = cond_2 \circ f_T$
2. Place preserving condition: $\bullet(f_S(term, p)) = f_T(\bullet(term, p))$ and
   $(f_S(term, p))\bullet = f_T((term, p)\bullet)$ for all $p \in P_1$ and $term \in T_{OP_1}(X_1)$
   where $\bullet(term, p) = \{t | (term, p) \leq post(t)\}$ and
   $(term, p)\bullet = \{t | (term, p) \leq pre(t)\}$ define the pre and post sets of p.
3. f_T, f_P and f_{SPEC} are injective and f_{SPEC} is persistent (compare [EM85]).
4. Embedding condition: $f_S(pre_1(t)) \leq pre_2(f_T(t))$ and
   $f_S(post_1(t)) \leq post_2(f_T(t))$ for all $t \in T_1$
5. $f_A : A_1 \xrightarrow{\sim} V_{f_{SPEC}}(A_2)$ is an isomorphism in $\mathbf{Alg(SPEC_1)}$ △

Remark: Intuitively, the conditions ensure that

1. the firing conditions are preserved by the morphism,
2. arcs adjacent to places are not changed,
3. the morphism is an injection and the target specification is a correct extension of the source specification,
4. the morphism has to map all arcs,
5. the algebra is merely extended for the new parts of the target specification or it is merely renamed.

Note the difference between place preserving morphisms (def. 3.1) and the (transition preserving) AHL net morphisms as defined in def. 2.2. The commutativity of diagram (**2**) in def. 2.2 yields a preservation of transitions in the sense that no new arcs are added to mapped transitions and no old (mapped) arcs are deleted from their pre and post domains. Place preserving morphisms are in general not transition preserving because condition 4 in def. 3.1 expresses that the pre and post domain of a transition in N_2 may contain more places than the original transition in N_1. A morphism $f : N_1 \rightarrow N_2$ that is place preserving and transition preserving at the same time merely yields a disjoint embedding of N_1 into N_2.

Example 3.2 (Place Preserving Morphism in Hypertension Test)
We sketch that the morphism $f^{\mathsf{VVM}} : L \rightarrow R$ determined by figure 2 is place preserving. The inclusions $f^{\mathsf{VVM}}{}_P$ and $f^{\mathsf{VVM}}{}_T$ are given implicitly using name identity. The specification morphism $f^{\mathsf{VVM}}{}_{SPEC}$ is an inclusion as sorts, operations and equations concerning the hypertension test are added in $\underline{\underline{\mathsf{VVM\text{-}Spec}_{hyper}}}$ such that $f^{\mathsf{VVM}}{}_{SPEC}$ is persistent (see [PGE97]). The conditions of def. 3.1 hold such that the morphism $f^{\mathsf{VVM}} : L \rightarrow R$ is place preserving:
Condition 1 is satisfied because transitions in net R that lie in the image of f^{VVM} have the same firing conditions as their originals in net L. Condition 2 is satisfied as no new arcs are adjacent to mapped places. For the place **vital value taken** this is formally shown, for the other places it is analogous:

$$\bullet(f^{\mathsf{VVM}}{}_S(v(BPd, BPs, T, PatId), \textbf{vital value taken}))$$
$$= \quad \{\textit{\textbf{Taking blood pressure}}\}$$
$$= \quad f^{\mathsf{VVM}}{}_T(\{\textit{\textbf{Taking blood pressure}}\})$$
$$= \quad f^{\mathsf{VVM}}{}_T(\bullet(v(BPd, BPs, T, PatId), \textbf{vital value taken}))$$

analogously $\quad (f^{\mathsf{VVM}}{}_S(V, \textbf{vital value taken}))\bullet = f^{\mathsf{VVM}}{}_T((V, \textbf{vital value taken})\bullet)$
Moreover, $f^{\mathsf{VVM}} : L \rightarrow R$ is an embedding (condition 4) as no arcs are deleted. The morphism f^{VVM} is not transition preserving in the sense of def. 2.2 because the transition ***Taking blood pressure*** in R has more places in its post set than the original ***Taking blood pressure*** in L. ◇

We will now define formulas over AHL net markings and their translations via morphisms to be able to express safety properties and prove their preservation via morphisms in a formal way.
The invariant formula $\Box\varphi$ expresses safety properties in the sense of [MP92]. Note that we use a restricted notion as φ is merely a static formula whereas in [MP92] backward operators are allowed.

Definition 3.3 (Formulas and Translations)
Let N be an AHL net according to definition 2.2. We define

- **Formulas**: For $\lambda \in \mathbb{N}$ and $(a,p) \in (A \times P)$: $\lambda(a,p)$ is a **static formula**. For φ_1, φ_2 static formulas: $\neg\varphi_1, \varphi_1 \wedge \varphi_2$ are static formulas. Let φ be a static formula over N. Then $\Box\varphi$ is an **invariant formula**.
- **Validity of formulas:** Let $M \in (A \times P)^{\oplus}$ be a marking and let φ_1 and φ_2 be static formulas. A static formula under the marking M is valid if:

$$\begin{aligned} M \models_N \varphi_1 &\iff \varphi_1 \leq M \qquad \textit{for } \varphi_1 = \lambda(a,p) \\ M \models_N \neg\varphi_1 &\iff \neg(M \models_N \varphi_1) \\ M \models_N \varphi_1 \wedge \varphi_2 &\iff (M \models_N \varphi_1) \wedge (M \models_N \varphi_2) \end{aligned}$$

  The invariant formula $\Box\varphi$ holds in N under M iff φ holds in all states reachable from M: $M \models_N \Box\varphi \iff \forall M' \in [M >: M' \models_N \varphi$
- **Translation of formulas**: Let $f = (f_{SPEC}, f_P, f_T, f_A) : N_1 \to N_2$ be a place preserving AHL net morphism. Then the translation $\mathcal{T}_f$ of formulas over N_1 under the marking $M_1 \in (A_1 \times P_1)^{\oplus}$ to formulas over N_2 is given as follows, where f_M is defined as in def. 2.2:

$$\begin{aligned} \mathcal{T}_f(\varphi) &= f_M(\varphi) \qquad \text{for } \varphi = \lambda(a,p) \in (A_1 \times P_1)^{\oplus} \\ \mathcal{T}_f(\neg\varphi) &= \neg\mathcal{T}_f(\varphi) \\ \mathcal{T}_f(\varphi_1 \wedge \varphi_2) &= \mathcal{T}_f(\varphi_1) \wedge \mathcal{T}_f(\varphi_2) \\ \mathcal{T}_f(\Box\varphi) &= \Box\mathcal{T}_f(\varphi) \end{aligned}$$

△

Next, we explain how a translated formula $\mathcal{T}_f(\varphi)$ is evaluated under a translated marking $M_2 \in (A_2 \times P_2)^{\oplus}$. Let us define the notion of a translated marking M_2 via the notion of a restriction of the marking M_2 with respect to f as we are only interested in the marked places of M_2 that are images of places of N_1.

Definition 3.4 (Restriction of Marking)
Let $f : N_1 \to N_2$ be a place preserving AHL net morphism, $M_1 \in (A_1 \times P_1)^{\oplus}$ a marking of N_1 and $M_2 \in (A_2 \times P_2)^{\oplus}$ a marking of N_2
s.t. $M_2 = f_M(M_1) \oplus \sum_{j=1}^{m} \mu_j(a_j, p_j)$ with $\mu_j(a_j, p_j) \notin f_M(A_1 \times P_1)^{\oplus}$
Then the **restriction** $M_{2|f}$ of the marking M_2 to the net N_1 with respect to f is given as follows: $M_{2|f} := M_1$ △

$M_{2|f}$ is well-defined due to the injectivity of the underlying morphisms.

Now we come to the main theorem concerning the preservation of formulas by morphisms.

Theorem 3.5 (Place Preserving Morphisms Preserve Safety Properties)
Let $f : N_1 \to N_2$ be a place preserving AHL net morphism and $M_1 \in (A_1 \times P_1)^{\oplus}$ and $M_2 \in (A_2 \times P_2)^{\oplus}$ be markings of N_1 and N_2 with $M_{2|f} = M_1$. Let $\Box\varphi$ be an invariant formula. Then the following holds:

$$M_1 \models_{N_1} \Box\varphi \Longrightarrow M_2 \models_{N_2} \mathcal{T}_f(\Box\varphi)$$

△

Example 3.6 (Preserving a Safety Property in Hypertension Test)
As example we consider the place preserving morphism $f^{\text{VVM}} : L \to R$ as given in ex. 3.2. Assume we have a marking M_1 in L analogously to ex. 2.1 and 2.4. Then we have $M_1 \models_L \varphi^{\text{VVM}}$ due to the same argument as in ex. 2.1. As we have a place preserving morphism (see ex. 3.2), we can apply theorem 3.5. Note that marking and safety property do not change, as f^{VVM} is an inclusion. Thus, we have $M_1 \models_L \varphi^{\text{VVM}}$ implies $M_2 \models_R \varphi^{\text{VVM}}$, where M_2 contains M_1. $\diamond$

Proof Idea of Theorem 3.5
For a complete proof we refer to the detailed technical report [PGE97].
The proof takes four steps: First the effect of restriction to the pre and post domains of transitions is investigated. Then we can show that the follower marking is preserved. Thirdly we show the preservation of static formulas. At last we prove the preservation of invariant formulas.

Let N_1, N_2 be AHL nets and $f : N_1 \to N_2$ a place preserving AHL net morphism. Let $M_1 \in (A_1 \times P_1)^{\oplus}$ be a marking of N_1 and $M_2 \in (A_2 \times P_2)^{\oplus}$ be a marking of N_2 with $M_{2|f} = M_1$.

Restriction and Place Preserving Morphisms $(*)$

(1) For all $t_2 \in T_2$ with $f_T(t_1) = t_2$:
  (i) $pre_2(t_2)_{|f} = pre_1(t_1)$ and (ii) $post_2(t_2)_{|f} = post_1(t_1)$
(2) For all $t_2 \in T_2 \backslash f_T(T_1)$:
  (i) $pre_2(t_2)_{|f} = \epsilon$ and (ii) $post_2(t_2)_{|f} = \epsilon$

For the proof of (1)(i) we show $pre_2(f_T(t_1))_{|f} \geq pre_1(t_1)$ directly, due to the embedding condition and $pre_2(f_T(t_1))_{|f} \leq pre_1(t_1)$ by contradiction, using the place preserving condition. The proof of (1)(ii) is analogous.

For the proof of (2)(i) assume $pre_2(t_2)_{|f} \not= \epsilon$. Let $(term_2, p_2) \leq pre_2(t_2)$ with $f_S(term_1, p_1) = (term_2, p2)$. Then $t_2 \in (term_2, p2)\bullet$ implies $t_2 \in (f_S(term_1, p_1))\bullet$. Thus f place preserving (see def. 3.1) implies $t_2 \in f_T((term_1, p_1)\bullet)$. This contradicts to our assumption $t_2 \in T_2 \backslash f_T(T_1)$. Hence $pre_2(t_2)_{|f} = \epsilon$. The proof of (2)(ii) is analogous.

Preservation of Follower Marking $(**)$

We prove $\forall M_2' \in [M_2 >$: $M_{2|f}' \in [M_1 >$ by induction over the firing of every transition in N_2 beginning with the induction base that no transition has been fired. As induction step we show that $M_2'[t_2, \overline{asg_2} > M_2''$ implies $M_{2|f}'' \in [M_1 >$.
For $t_2 \in T_2 \backslash f_T(T_1)$ we show with the help of $(*)(2)$ that $M_{2|f}'' = M_{2|f}' \in [M_1 >$.
For $t_2 = f_T(t_1)$ we first show that t_1 is enabled under $\overline{asg_1}$ and $M_{2|f}'$ with $\overline{asg_1} = X_1 \longrightarrow X_2 \xrightarrow{asg_2} A_2 \xrightarrow{iso} A_1$. With t_1 enabled we can show that $M_{2|f}'[t_1, \overline{asg_1} > M_{2|f}''$. Together with $M_{2|f}' \in [M_1 >$ we have: $M_{2|f}'' \in [M_1 >$.

Preservation of Static Formulas $(***)$

We show that $M_1 \models_{N_1} \varphi \iff M_2 \models_{N_2} \mathcal{T}_f(\varphi)$ by induction over the structure of static formulas as given in def. 3.3.

Preservation of Invariant Formulas
We show $M_1 \models_{N_1} \Box\varphi$ implies $M_2 \models_{N_2} \mathcal{T}_f(\Box\varphi)$ by

$$\begin{aligned}
M_1 \models_{N_1} \Box\varphi &\Longleftrightarrow \forall M_1' \in [M_1 >: M_1' \models_{N_1} \varphi && \text{due to def. 3.3}\\
&\Longrightarrow \forall M_2' \in [M_2 >: M'_{2|f} \models_{N_1} \varphi && \text{due to } (**)\\
&\Longleftrightarrow \forall M_2' \in [M_2 >: M_2' \models_{N_2} \mathcal{T}_f(\varphi) && \text{due to } (***)\\
&\Longleftrightarrow M_2 \models_{N_2} \Box\mathcal{T}_f(\varphi) && \text{due to def. 3.3}\\
&\Longleftrightarrow M_2 \models_{N_2} \mathcal{T}_f(\Box\varphi) && \surd
\end{aligned}$$

4 Rule-Based Refinement Preserving Safety Properties

In software engineering it is most desirable to "automatically" derive properties of a refined net from its abstraction. The main advantage of such refinements is that those properties do not have to be proven again. For our example in section 2 it is obvious that the safety property φ^{VVM} *"At any time we have: there is some patient at the ward if and only if the corresponding patient record is at the ward."* should hold again in the resulting net after the application of a rule r^{VVM} to a net with that property. This is captured by the notion of 'rule-based refinement', which is rule-based modification plus the preservation of (safety) properties. A main advantage of rules is that modifications are described locally, i.e. in a small context. Moreover — as we will show in this section — the preservation of safety properties can already be checked on the level of rules. Applying these rules means an intrinsic propagation of safety properties to the resulting net. This, of course, is of high importance from a software engineering point of view, especially for our case study in section 2 because there is no need of verifying these properties in the resulting net.

Concept 4.1 (Vertical Structuring Technique: Rule-Based Refinement)

Rule-based refinement is the extension of rule-based modification with morphisms that preserve system properties. In this case we have place preserving morphisms that preserve safety properties. A safety property preserving rule in rule-based refinement is given by (r, f) with a rule $r : L \leftarrow K \rightarrow R$ (see def. 2.3) and $f : L \rightarrow R$ a place preserving morphism (see def. 3.1) which are compatible in the sense that the composition of $K \rightarrow L$ and $L \rightarrow R$ equals $K \rightarrow R$, i.e. $K \rightarrow L \rightarrow R = K \rightarrow R$. ◊

The following theorem provides sufficient conditions for propagating safety properties from one net to its modification. The general idea is that the application of a rule that preserves safety properties leads to a net transformation that preserves the same safety properties. In fact, these rules that have place preserving morphisms from the left to the right hand side, preserve safety properties.

Theorem 4.2 (Rule-Based Refinement Preserves Safety Properties)
Let (r, f) be a safety property preserving rule (see con. 4.1), $N_1 \overset{r}{\Longrightarrow} N_2$ the application of r to the net N_1. Furthermore let M_1 be a marking of N_1 and M_2

a marking of N_2 with $M_2|_f = M_1$ (see def. 3.4).
Then there is a well-defined morphism $\overline{f} : N_1 \rightarrow N_2$ induced by $f : L \rightarrow R$ with:

$$M_1 \models_{N_1} \Box\varphi \quad \Longrightarrow \quad M_2 \models_{N_2} \mathcal{T}_{\overline{f}}(\Box\varphi)$$

△

Example 4.3 (Refinement in Hypertension Test)
Again we consider the example given in figure 2 with respect to the safety property φ^{VVM} *"At any time we have: there is some patient at the ward if and only if the corresponding patient record is at the ward."* It has already been shown that $f^{\mathsf{VVM}} : L \rightarrow R$ is place preserving (see ex. 3.2) and the above safety property holds in VVM with marking M^{VVM} (see ex. 2.1). Furthermore we have $K \rightarrow L \rightarrow . R = K \rightarrow R$ as all morphisms are inclusions. Thus, due to theorem 4.2 for any M_2 with $M_2|_{\overline{f^{\mathsf{VVM}}}} = M^{\mathsf{VVM}}$ we have: $M_2 \models_{\mathsf{BEX}} \varphi^{\mathsf{VVM}}$
Note, $\varphi^{\mathsf{VVM}} = \mathcal{T}_{\overline{f^{\mathsf{VVM}}}}(\varphi^{\mathsf{VVM}})$ as $\overline{f^{\mathsf{VVM}}}$: VVM → BEX is also an inclusion. ◇

In order to prove theorem 4.2 we apply results of [Pad96] that are formulated for an arbitrary category and a distinguished class $\mathcal{Q}$ of morphisms used to classify different types of rules. In this case the class $\mathcal{Q}$ is given by the place preserving morphisms, those preserving safety properties as shown in theorem 3.5.
Technically, we must prove the assumptions given in [Pad96], def. 4.3.1. We therefore define a category (**QAHL**) that contains both the category **AHL** (see def. 2.2) and place preserving morphisms. In this category pushouts of the subcategory **AHL** must be preserved, and the class of place preserving morphisms must be closed under pushouts and coproducts. Note, that we only sketch the proof for restrictions of space. They can all be found in detail in [PGE97].
Proof Idea of Theorem 4.2:
We first present the general definition of $\mathcal{Q}$-morphisms according to [Pad96], (def. 4.3.1).

> *Let* **QCAT** *be a category so that* **CAT** *is a subcategory* **CAT** $\subseteq$ **QCAT** *and the inclusion functor* I : **CAT** → **QCAT** *preserves pushouts.*
> *Let* $\mathcal{Q}$ *be a class of morphisms in* **QCAT**, *closed under the construction*
>
> - *of pushouts in* **QCAT**: *Given* $C \xrightarrow{f'} D \xleftarrow{g'}$ *a pushout of* $B \xleftarrow{f} A \xrightarrow{g} C$, *then* $f \in \mathcal{Q} \implies f' \in \mathcal{Q}$.
> - *of coproducts in* **QCAT**: *For* $A \xrightarrow{f} B$ *and* $A' \xrightarrow{f'} B'$, *we have* $f, f' \in \mathcal{Q} \implies f + f' \in \mathcal{Q}$ *provided the coproduct* $A + A' \xrightarrow{f+f'} B + B'$ *of* f *and* f' *exists in* **QCAT**.

We instantiate this definition with algebraic high-level nets, that is **AHL** and **QAHL** correspond to **CAT**, resp. **QCAT**.
Category QAHL: Objects in **QAHL** are AHL nets as defined in def. 2.2.
Given two AHL nets $N_i = (SPEC_i, P_i, T_i, pre_i, post_i, cond_i, A_i)$ for $i = 1, 2$, a morphism $f : N_1 \rightarrow N_2$ is a quadrupel $f = (f_{SPEC}, f_P, f_T, f_A)$, where the components are morphisms in the underlying categories **SPEC** and **SETS** and f_E, f_S are defined as in def. 2.2 , satisfying the following conditions:

(i) $f_E \circ cond_1 = cond_2 \circ f_T$

(ii) arcs are preserved, i.e. $\forall t_1 \in T_1$:

(a) $f_S(pre_1(t_1)) \leq pre_2(f_T(t_1))$

(b) $f_S(post_1(t_1)) \leq post_2(f_T(t_1))$

(iii) $f_A : A_1 \xrightarrow{\sim} V_{f_{SPEC}}(A_2)$ is an isomorphism in the category **Alg(SPEC$_1$)** of $SPEC_1$-algebras.

Preservation of Pushouts: The proof uses the fact that pushouts in **AHL** are constructed componentwise in **SPEC**, **SETS** and **Alg(SPEC)** (see e.g. fact 5.2 in [PER95]). Thus, there are unique induced morphisms in these categories, whose combination is a unique **QAHL**-morphism.

Class $\mathcal{Q}$: The class $\mathcal{Q}$ of morphisms is given by the place preserving morphisms (see def. 3.1).

Preservation of $\mathcal{Q}$ under Pushouts: We have to show that the induced pushout morphisms satisfy the conditions of definition 3.1. The preservation of firing conditions (condition 1), the embedding condition (condition 4) and condition 5 are due to the notion of **QAHL**-morphisms. Condition 3 immediately follows by preservation of monomorphisms and the extension lemma 8.15 in [EM85]. Condition 2 (place preserving condition) can be shown by the mutual inclusion of sets. In one direction we use the linearity of free commutative monoids. For the other inclusion it is shown that an arc, decorated by an image term, between a transition and an image place in the pushout object must already exist in the source net.

Preservation of $\mathcal{Q}$ in Coproducts: We show, that coproducts in **QAHL** are constructed componentwise and the inclusions are place preserving. The actual proof is straightforward and uses mainly inclusions.

These conditions and fact 4.3.3 in [Pad96] yield: $N_1 \overset{r}{\Longrightarrow} N_2$ and $f : L \to R$ in $\mathcal{Q}$ implies a well-defined induced morphism $\overline{f} : N1 \to N2$ in $\mathcal{Q}$. This and theorem 3.5 proves the stated fact. $\surd$

Implication for Formal Software Development and Open Problems

As illustrated above rule-based refinement allows deducing safety properties of the refined net under very weak assumptions. We have shown that the preservation of safety properties can be expressed in terms of the transforming rule and the source net. Only the source net has to satisfy this property (under a marking) and the rule has to be place preserving. In so far verification of these properties in a target net can exploit the properties of the source net, which seems to be very natural in the context of system development. Thus, iterative verification becomes possible: the safety property has to be verified once for a starting net and from there on the safety property is propagated by place preserving rules.

A further important aspect is the relation between vertical, i.e. rule-based refinement, and horizontal structuring realized by the notion of union and fusion. Intuitively, union is the gluing of two nets sharing a common subnet. It serves the purpose of joining two parts over a common interface. Fusion is the identification of distinct items in one net, which means unification or abstraction. In fact, the general theory of $\mathcal{Q}$-morphisms developed in [Pad96] states the compatibility of rule-based refinement with horizontal structuring (see [Pad96], theorem

4.5.5 and 4.5.9). This is a fundamental issue from the software engineering point of view as it allows concurrent refinement and composition of the system. The following diagram illustrates this fact:

$$\begin{tikzcd}
N_1 \arrow[rr, Rightarrow] \arrow[rr, dotted, bend left, no head] \arrow[dd, no head] & & N_1' \arrow[dd, no head] & \\
& N_2 \arrow[rr, Rightarrow] \arrow[rr, dotted, bend left, no head] \arrow[dl, no head] & & N_2' \arrow[dl, no head] \\
N \arrow[rr, Rightarrow] \arrow[rr, dotted, bend left, no head] & & N' &
\end{tikzcd}$$

The preservation of safety properties, which is supplied on the level of rules, cannot only be transferred to transformations but also to the horizontal composition of transformations. This means, that we can do horizontal structuring (denoted by the forking lines) and rule-based refinement (denoted by the double arrows) in any ordering.

From a software engineering point of view an additional compatibility is essential, namely the propagation of safety properties from a component net (N_1) to the composed net (N). By transitivity this would ensure, that a safety property of a component net would be preserved in the refined and composed net (N'). This has been subject to our paper [EG97], where we show that propagation of safety properties is possible in special cases. However, a corresponding general theory still has to be developed.

5 Conclusion and Future Research

We have presented a new, formal vertical structuring technique, the rule-based refinement of algebraic high-level nets. This technique combines the advantages of a rule-based approach to stepwise development with a refinement that preserves safety properties. We have shown that a specific kind of AHL morphisms, called place preserving, allows transfering safety properties from the source to the target net. In combination with rule-based modification we obtain rule-based refinement – based on the theory of $\mathcal{Q}$-transformations in [Pad96] – that preserves safety properties. We have illustated this rule-based refinement for one refinement step using our main theorem 4.2 to transfer an important safety property in the context of our case study HDMS. This transfer is guaranteed in terms of the corresponding rule, and has not to be done for the whole subsystem. Moreover, this new technique of rule-based refinement can be adapted easily to other high-level Petri net formalisms, because in [Pad96, EP97] a uniform approach to Petri nets is developed where the abstract frame for rule-based refinement is already formulated.

Further research, aside from the compatibility of safety properties with horizontal structuring as discussed in section 4, concerns the transfer of other system properties as liveness, obligations, persistency and others as in [MP92] along the rules within the frame of rule-based refinement.

Moreover, as one of our referees suggested, the extension of our approach to different notions of time is also an exciting idea. This would lead to the notion of time preserving morphisms and rules which could be very helpful for the specification of real time systems.

References

[BDH92] E. Best, R. Devillers, and J. Hall. The Box Calculus: a new causal algebra with multi-label communication. In *Advances in Petri Nets*, pages 21–69. Lecture Notes in Computer Science, 1992. 609.

[BGV90] W. Brauer, R. Gold, and W. Vogler. A Survey of Behaviour and Equivalence Preserving Refinements of Petri Nets. *Advances in Petri Nets, LNCS 483*, 1990.

[BS90] J. Bradfield and C. Stirling. Verifying temporal properties of processes. In J. C. M. Baeten et al., editors, *LNCS; CONCUR'90, Theories of Concurrency: Unification and Extension. (Conference, 1990, Amsterdam, The Netherlands)*, pages 115–125, Springer, Berlin, 1990.

[CHL95] F. Cornelius, H. Hußmann, and M. Löwe. The KORSO Case Study for Software Engineering with Formal Methods: A Medical Information System. In M. Broy and S. Jähnichen, editors, *KORSO: Methods, Languages, and Tools for the Construction of Correct Software*, pages 417–445. Springer LNCS 1009, 1995. Also appeared as technical report 94-5, TU Berlin.

[DDGJ90] W. Damm, G. Döhmen, V. Gerstner, and B. Josko. Modular verification of petri nets: The temporal logic approach. In J. W. de Bakker et al., editors, *LNCS; Proceedings of the REX Workshop on Stepwise Refinement, 1989, Mook, The Netherlands*, pages 180–207, Springer, Berlin, 1990.

[DHP91] C. Dimitrovici, U. Hummert, and L. Petrucci. Composition and net properties of algebraic high-level nets. In *Advances of Petri Nets*. Springer Verlag Berlin LNCS 524, 1991.

[DM90] J. Desel and A. Meceron. Vincinity Respecting Net Morphisms. In *Advances in Petri Nets*, pages 165–185. Springer Verlag LNCS 483, 1990.

[EG97] C. Ermel and M. Gajewsky. Expanding the Use Of Structuring: Formal Justification for Working on Subnets. In Proceedings of *Workshop Petri Nets in System Engineering '97*, pages 44 – 54, University Hamburg, 1997. FBI -HH-B-205/97.

[EHKP91] H. Ehrig, A. Habel, H.-J. Kreowski, and F. Parisi-Presicce. Parallelism and concurrency in high-level replacement systems. *Math. Struct. in Comp. Science*, 1:361–404, 1991.

[Ehr79] H. Ehrig. Introduction to the algebraic theory of graph grammars. In V. Claus, H. Ehrig, and G. Rozenberg, editors, *1st Graph Grammar Workshop, LNCS 73*, pages 1–69. Springer Verlag, 1979.

[EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1: Equations and Initial Semantics*, volume 6 of *EATCS Monographs on Theoretical Computer Science*. Springer Verlag, Berlin, 1985.

[EP97] H. Ehrig and J. Padberg. Introduction to Universal Parametrized Net Classes. In H. Weber, H. Ehrig, and W. Reisig, editors, *MoveOn-Proc. der DFG-Forschergruppe "Petrinetz-Technologie"*, Technical Report 97-21, TU Berlin, 1997.

[EPE96] C. Ermel, J. Padberg, and H. Ehrig. Requirements Engineering of a Medical Information System Using Rule-Based Refinement of Petri Nets. In D. Cooke, B.J. Krämer, P. C-Y. Sheu, J.P. Tsai, and R. Mittermeir, editors, *Proc. Integrated Design and Process Technology*, pages 186 – 193. Society for Design and Process Science, 1996. Vol.1.

[Erm96] C. Ermel. Anforderungsanalyse eines medizinischen Informationssystems mit Algebraischen High-Level-Netzen. Techn. Report 96-15, TU Berlin, 1996.

[Gen91] H.J. Genrich. Predicate/Transition Nets. In *High-Level Petri Nets: Theory and Application*, pages 3–43. Springer, 1991.

[GG90] R.J. van Glabbeck and U. Golz. Equivalences and Refinement. In *Semantics of Systems of Concurrent Processes*, pages 309–333. Springer, 1990. Lecture Notes in Computer Science 469.

[GL81] H.J. Genrich and K. Lautenbach. System modelling with high-level Petri nets. *Theorĕtical Computer Science*, 13:109–136, 1981.

[HRH91] R. R. Howell, L. E. Rosier, and Chun Yen Hsu. A taxonomy of fairness and temporal logic problems for petri nets. *Theoretical Computer Science*, 82(2):341–372, 1991.

[Hum89] U. Hummert. *Algebraische High-Level Netze.* PhD thesis, Technische Universität Berlin, 1989.

[JCHH91] K. Jensen, S. Christensen, P. Huber, and M. Holla. *Design/CPN. A Reference Manual.* Meta Software Cooperation, 125 Cambridge Park Drive, Cambridge Ma 02140, USA, 1991.

[Jen92] K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, volume 1. Springer, 1992.

[Jen95] K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, volume 2. Springer, 1995.

[Lil94] J. Lilius. *On the Structure of High-Level Nets.* PhD thesis, Helsinki University of Technology, 1994.

[MP92] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems, Specification.* Springer-Verlag, 1992.

[OSS94] A. Oberweis, G. Scherrer, and W. Stucky. INCOME/STAR: Methodology and Tools for the Development of Distributed Information Systems. *Information Systems*, 19(8):643–660, 1994.

[Pad96] J. Padberg. *Abstract Petri Nets: A Uniform Approach and Rule-Based Refinement.* PhD thesis, Technical University Berlin, 1996. Shaker Verlag.

[PER95] J. Padberg, H. Ehrig, and L. Ribeiro. Algebraic high-level net transformation systems. *Math. Struct. in Computer Science*, 5:217–256, 1995.

[Peu97] S. Peuker. Invariant property preserving extensions of elementary petri nets. *Technical Report No.97-21, TU Berlin*, 1997.

[PGE97] J. Padberg, M. Gajewsky, and C. Ermel. Refinement versus Verification: Compatibility of Net-Invariants and Stepwise Development of High-Level Petri Nets. Technical Report 97–22, TU Berlin, 1997. to appear.

[Rei91] W. Reisig. Petri Nets and Algebraic Specifications. *Theoretical Computer Science*, 80:1–34, 1991.

[Sch96] K. Schmidt. *Symbolische Analysemethoden für algebraische Petri-Netze*, volume 4. Bertz Verlag, versal edition, 1996.

[SM97] J. Svensson and M. Meier. *Handbuch LEU Support-Guide.* Vebacom Service GmbH. Also, http://www.lion.de/PRODUKT/produkt.html.

[Vau87] Vautherin, J. Parallel System Specification with Coloured Petri Nets. In Rozenberg, G., editor, *Advances in Petri Nets 87*, pages 293–308. LNCS 266, Springer, 1987.