Relating Different Semantics for Object Petri Nets.

Formal Proofs and Examples.

Rüdiger Valk.

Technical Report FBI-HH-B-226, University of Hamburg, Department for Computer Science Report/00, 2000.

Universität Hamburg
Fachbereich Informatik
Vogt-Kölln-Str. 30
D-22527 Hamburg

Bericht 226

Relating Different Semantics for Object Petri Nets
Formal Proofs and Examples

FBI-HH-B-226/00

Rüdiger Valk
Universität Hamburg
Fachbereich Informatik
valk@informatik.uni-hamburg.de

Abstract

Object Nets belong to a class of Petri nets allowing for a two-level way of modeling by giving tokens of a Petri net the structure of a Petri net again. The usefulness of this approach has been shown in numerous case studies, ranging from modeling distributed algorithms to workflow and flexible manufacturing systems. It allows for the modeling of real world objects by tokens having their own dynamical behavior. As it is well-known from the field of distributed systems in general, (at least) two different ways of object management are of interest when implementing remote access: either by referencing to a single representation or by creating copies which are treated in a consistent way. In analogy to programming language constructs, this is denoted by reference and value semantics, respectively. In this contribution value and reference semantics of object nets are formally defined. Conditions are presented that allow the transfer from one of these semantics to the other. While the proof techniques strongly rely on partial orders (causal nets), the comparison is made on the basis of occurrence sequences, which serve as a common description language.

Zusammenfassung

Objektnetze gehören zu einer Klasse von Petrinetzen, die eine zweistufige Modellierungstechnik dadurch unterstützen, dass die Marken selbst wieder Petrinetze sein dürfen. Dies hat sich bereits in umfangreicher Modellierung für Workflow- und flexible Fertigungs-Systeme als sehr vorteilhaft erwiesen. Objekte der realen Welt wie Aufträge oder Fertigungsteile werden dabei mit ihrem Bearbeitungsplan und ihrer Fortentwicklung dargestellt. Im Kontext verteilter Systeme sind als Zugriffsart für nichtlokale Objekte bekanntlich (mindestens) zwei Formen von Interesse: die Referenz auf eine einzige zentrale Objektrepräsentation oder die Erzeugung und konsistente Pflege von Kopien in unterschiedlichen Systemteilen. In Analogie zur Programmiersprachenterminologie werden diese als Referenz- bzw. Wert-Semantik bezeichnet. In diesem Bericht werden Referenz- und Wert-Semantik von Objektnetzen formal eingeführt. Außerdem werden Bedingungen für einen konsistenten Wechsel zwischen ihnen bewiesen. Während die Beweise intensiv von partiellen Ordnungen (Kausalnetzen) Gebrauch machen, basiert der Vergleich der Semantiken auf Ausführungsfolgen.

Contents

List of Figures

1 Introduction

The class of object nets has been shown to be very useful in modeling application systems using the object oriented modeling paradigm. Object-oriented modeling means that software is designed as the interaction of discrete objects, incorporating both data structure and behavior [Ra91]. From a Petri net point of view dynamical objects are modeled as nets which are token objects in a general system Petri net. They can be seen as tokens of a particular form of high level nets. In contrast to Colored Nets, however, the tokens have the structure of a net again. We therefore distinguish the base nets, called *System Nets* or *Environment Nets* from its tokens in net form, called *Object Nets* or *Token Nets*.

Object nets move through a system net like ordinary tokens. In the current formalism they are able to change their marking, but not their structure. The change of the object net marking can be independent from the system net. Such a step is called an *autonomous* occurrence of the transition. If this change is triggered by the system net, it is called *interaction*. Interaction is formalized as the incidental occurrence of a system net transition t and an object net transition e. Sometimes this form of synchronization is called a “rendez-vous”. In workflow-like applications interaction mostly means the execution of a subtask e by a functional unit t. In a different context it could mean that an operation is performed on an object. If there is no (internal) change of the object net marking, we call the effect a *transport*. This notion suggest an interpretation of mobile systems, but is clearly more general, as a topological structure of the system net is not necessarily assumed.

As in preceding papers we restrict both net types to elementary net systems and call them *Elementary Object Nets (EOS)*. They have been studied as *task systems* in earlier papers ([Val87a], [Val87b]). In ([Val98],[Val99b]) applications to the modeling of work flow and flexible manufacturing system are given. In the latter case the system net models the environment of machines, robots, conveyors etc. whereas the object nets are execution plans containing the current state of processing. The papers mentioned also contain a distributed version of the Five Philosophers Problem. Object nets in this example are philosophers that can enter or leave the dining room, and trolleys for requesting a missing fork from a neighbor. In [MV00] and [AMVW99] applications to business process modeling are given.

The difference of reference and value semantics for object nets may be informally described as follows. By some transition occurrence of an elementary object system a particular object net may appear in different places of the system net. In reference semantics these appearances consist of a reference to a single object net, whereas in value semantics these instances of the object net are considered as independent copies. The copies are identical in their net structure but may differ in their current marking.

Reference semantics has been formalized as *bi-markings* in [Val98] and [Val99b]. It is also the leading paradigm of *reference nets* as introduced in [Kum98] or as implemented in the *Renew* tool [KW99]. Value semantics was used when working with *process markings (p-markings)* in [Val98] and [Val99b]. Recently a different formalization using Linear Logic is given in [Far99]. The paper, presented here, is an extension of a report [Val99a] by elaborated proofs and new results.

The distinction between reference and value semantics corresponds to the distinction between local and distributed information in a distributed system and is therefor a central characteristics of communication based systems. Dynamic objects may be stored in a central way with respect to a node in a computer network (see Figure 1), but must be represented as consistent copies in different nodes. Concurrent work on separated copies may even be necessary in a single site as represented in the lower part of Figure 1.

In this paper we formally study differences and similarities of reference and value semantics for elementary object systems. This is motivated by the importance of the topic in general and by specific needs in practical work. To give an example for the latter, we refer to the workflow example of Figure 8 in [Val98]. The concurrent action of the transitions "*secretary*" and "*official 2*" is given there with respect to value semantics, but may be simulated with the Renew tool, which implements reference semantics. Hence, in this example the two semantics result in the same behavior. There are, however, numerous examples where different behavior is observed. The purpose of this paper is to discuss reasons for such differences and to give criteria for a similar behavior. To keep the presentation less complex we are restricted to the case where only one single object net exists. This model is called *Unary Elementary Object System*.

2 Unary Elementary Object Systems

In this section *Unary Elementary Object Systems* are introduced, consisting of a *system net SN* and an *object net ON*, both being elementary net systems. These are used in their standard form as given in [Thi87]. An *Elementary Net System (EN system)* $N = (B, E, F, C)$ is defined by a finite, non-empty set of *places* (or conditions) B, a finite, non-empty set of *transitions* (or events) E, disjoint from B, a flow relation $F \subseteq (B \times E) \cup (E \times B)$ and an *initial marking* (or initial

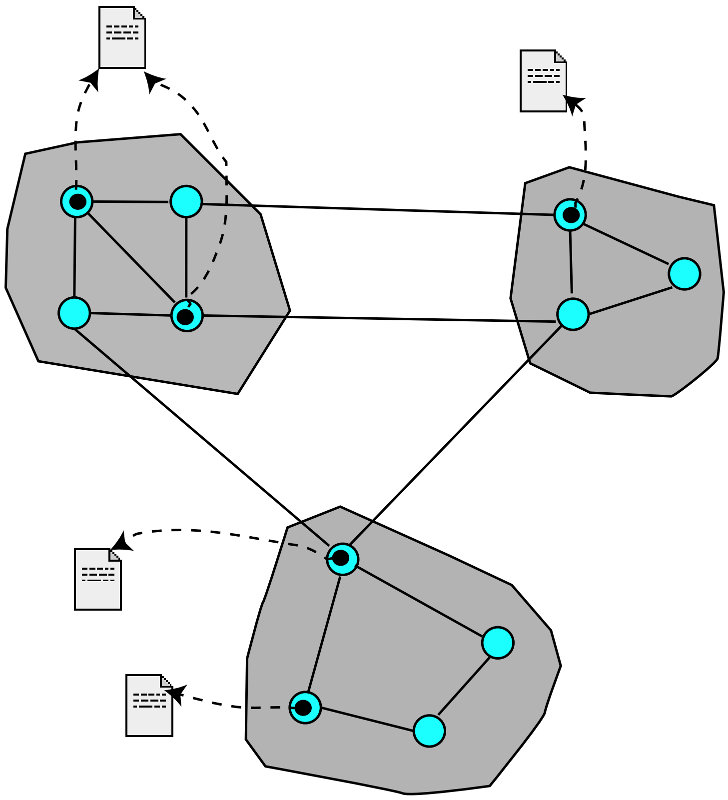

Figure 1: Central and distributed object reference

case) $C \subseteq B$. In addition to the usual condition $dom\,(F) \cup range(F) = B \cup E$ we assume $E \subseteq range(F)$, i.e. ${}^\bullet e \neq \emptyset$ for all $e \in E$. For the definition of ${}^\bullet e$, the domain *dom* and *range* of a relation see the appendix. The occurrence relation for markings C_1, C_2 and a transition t is written as $C_1 \rightarrow_t C_2$. If t is enabled in C_1 we write $C_1 \rightarrow_t$. These notions are extended to words $w \in E^*$, as usual, and written as $C_1 \rightarrow_w C_2$. $FS(N) := \{w \in E^* \mid C \rightarrow_w\}$ is the set of *firing* or *occurrence sequences* of N, and $R(N) := \{C_1 \mid \exists w : C \rightarrow_w C_1\}$ is the set of reachable markings (or cases), also called the *reachability set* of N (cf. [Roz87]). We will also use *processes* of EN systems in their standard definition [Roz87]. Their definition is given in the appendix.

Definition 2.1 *A* unary elementary object system *is a tuple* $EOS = (SN, ON, \rho)$ *where*

- $SN = (P, T, W, \mathbf{M}_0)$ *is an EN system, called* system net *of* EOS*,*
- $ON = (B, E, F, \mathbf{m}_0)$ *is an EN system, called* object net *of* EOS*, and*
- $\rho \subseteq T \times E$ *is the* interaction relation*.*

Throughout this paper the symbols P, T, W and B, E, F will be used to distinguish the system from the object net. Figure 2 gives an example of a unary elementary object system with the components of an object net ON on the left and a system net SN on the right. The interaction relation ρ is given by labels $< i_n >$ at t and e iff $(t, e) \in \rho$ ("i_n" stands for interaction number n, which has no other meaning apart from specifying interacting transitions).

2.1 Reference Semantics

Before proceeding to the formalization we describe the intuition behind the occurrence rule to be defined next. The token in the place p_1 of the system net in Figure 2 should be thought of as a reference to the object net ON. After the occurrence of the transport t_1 the places p_2 and p_4 are marked. These tokens can be seen as references to the same object net ON, which is still in its initial marking. Then two interactions $[t_2, e_2]$ and $[t_3, e_3]$ may occur leading to the marking $\mathbf{M} = \{p_3, p_5\}$. Both markings form references to the object net ON being now in the marking $\mathbf{m} = \{b_3, b_5\}$. Both markings form together the global marking of the EOS, which is called a *"bi-marking"* and will be denoted by $(\mathbf{M}, \mathbf{m})$. The successor bi-marking after the occurrence of $[t_4, e_4]$ is $(\mathbf{M}_1, \mathbf{m}_1)$ with $\mathbf{M}_1 = \{p_6\}$ and $\mathbf{m}_1 = \{b_6\}$.

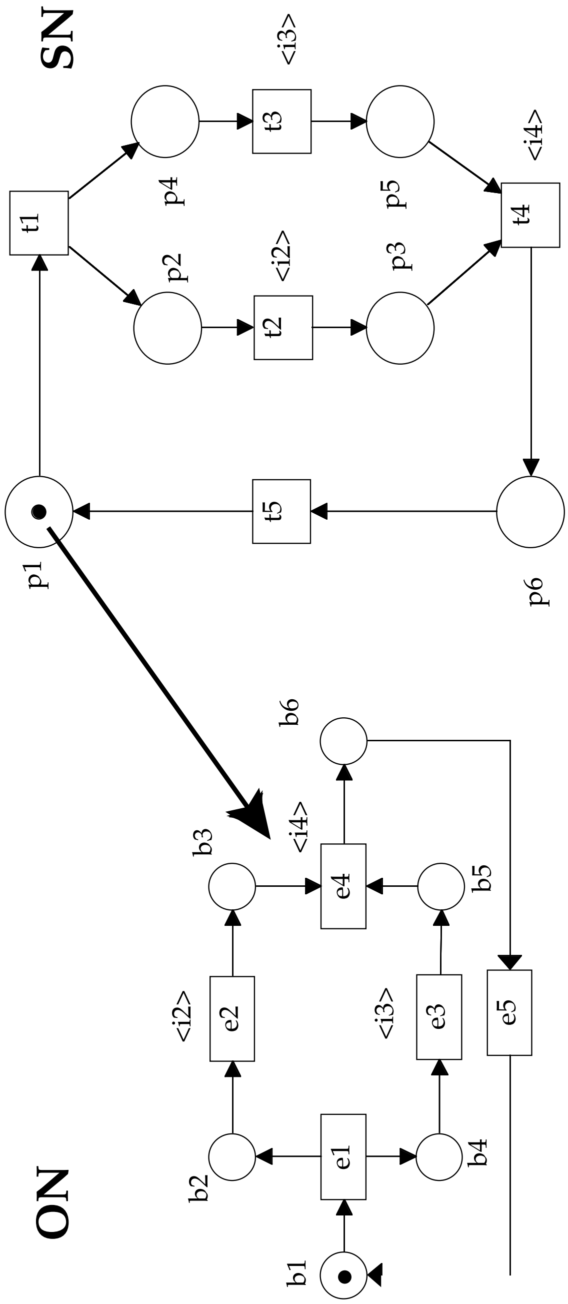

Figure 2: Elementary object system *con-task*

In the definitions of the occurrence rule we will use the following well-known notions for a binary relation ρ. For $t \in T$ and $e \in E$ let $t\rho := \{e \in E \mid (t,e) \in \rho\}$ and $\rho e := \{t \in T \mid (t,e) \in \rho\}$. Then $t\rho = \emptyset$ means that there is no element in the interaction relation with t.

Definition 2.2 *A* bi-marking *of a unary elementary object system* $EOS = (SN, ON, \rho)$ *is a pair* $(\mathbf{M}, \mathbf{m})$ *where* $\mathbf{M}$ *is a marking of the system net* SN *and* $\mathbf{m}$ *is a marking of the object net* ON*.*

a) A transition $t \in T$ *is* activated *in a bi-marking* $(\mathbf{M}, \mathbf{m})$ *of* EOS *if* $t\rho = \emptyset$ *and* t *is activated in* $\mathbf{M}$*. Then the successor bi-marking* $(\mathbf{M}', \mathbf{m}')$ *is defined by* $\mathbf{M} \xrightarrow{t} \mathbf{M}'$ *(w.r.t. SN) and* $\mathbf{m} = \mathbf{m}'$*. We write* $(\mathbf{M}, \mathbf{m}) \xrightarrow{[t,\lambda]} (\mathbf{M}', \mathbf{m}')$ *in this case.*

b) A pair $[t, e] \in T \times E$ *is* activated *in a bi-marking* $(\mathbf{M}, \mathbf{m})$ *of* EOS *if* $(t, e) \in \rho$ *and* t *and* e *are activated in* $\mathbf{M}$ *and* $\mathbf{m}$*, respectively. Then the successor bi-marking* $(\mathbf{M}', \mathbf{m}')$ *is defined by* $\mathbf{M} \rightarrow_t \mathbf{M}'$ *(w.r.t. SN) and* $\mathbf{m} \rightarrow_e \mathbf{m}'$ *(w.r.t. ON). We write* $(\mathbf{M}, \mathbf{m}) \xrightarrow{[t,e]} (\mathbf{M}', \mathbf{m}')$ *in this case.*

c) A transition $e \in E$ *is* activated *in a bi-marking* $(\mathbf{M}, \mathbf{m})$ *of* EOS *if* $\rho e = \emptyset$ *and* e *is activated in* $\mathbf{m}$*. Then the successor bi-marking* $(\mathbf{M}', \mathbf{m}')$ *is defined by* $\mathbf{m} \xrightarrow{e} \mathbf{m}'$ *(w.r.t. ON) and* $\mathbf{M}' = \mathbf{M}$*. We write* $(\mathbf{M}, \mathbf{m}) \xrightarrow{[\lambda,e]} (\mathbf{M}', \mathbf{m}')$ *in this case.*

In transition occurrences of type b) both the system and the object net participate in the same event. Such an occurrence is therefore called an *interaction*. By an occurrence of type c), however, the object net changes its state without moving to another place of the system net. It is therefore called *object-autonomous* or *autonomous* for short. The symmetric case in a) is called *system-autonomous* or *transport*, since the object net is transported to a different place without performing an action.

Definition 2.3 *The successor bi-marking relation* $(\mathbf{M}, \mathbf{m}) \xrightarrow{[\alpha,\beta]} (\mathbf{M}', \mathbf{m}')$ *is inductively extended to finite sequences* $\widetilde{w} \in \Gamma^*$ *(where* $\Gamma := (T \cup \{\lambda\}) \times (E \cup \{\lambda\})) \setminus [\lambda, \lambda]$ *and* $[\lambda, \lambda]$ *denotes the neutral element of the free monoid* Γ^**):*

- $(\mathbf{M}, \mathbf{m}) \xrightarrow{[\alpha,\beta]}_{ref} (\mathbf{M}, \mathbf{m})$ *if* $[\alpha, \beta] = [\lambda, \lambda]$ *and*
- $(\mathbf{M}, \mathbf{m}) \xrightarrow{\widetilde{w}[\alpha,\beta]}_{ref} (\mathbf{M}', \mathbf{m}')$ *if* $\exists (\mathbf{M}'', \mathbf{m}'')$. $(\mathbf{M}, \mathbf{m}) \xrightarrow{\widetilde{w}}_{ref} (\mathbf{M}'', \mathbf{m}'')$ *and* $(\mathbf{M}'', \mathbf{m}'') \xrightarrow{[\alpha,\beta]} (\mathbf{M}', \mathbf{m}')$ *for* $\widetilde{w} \in \Gamma^*$ *and* $[\alpha, \beta] \in \Gamma$

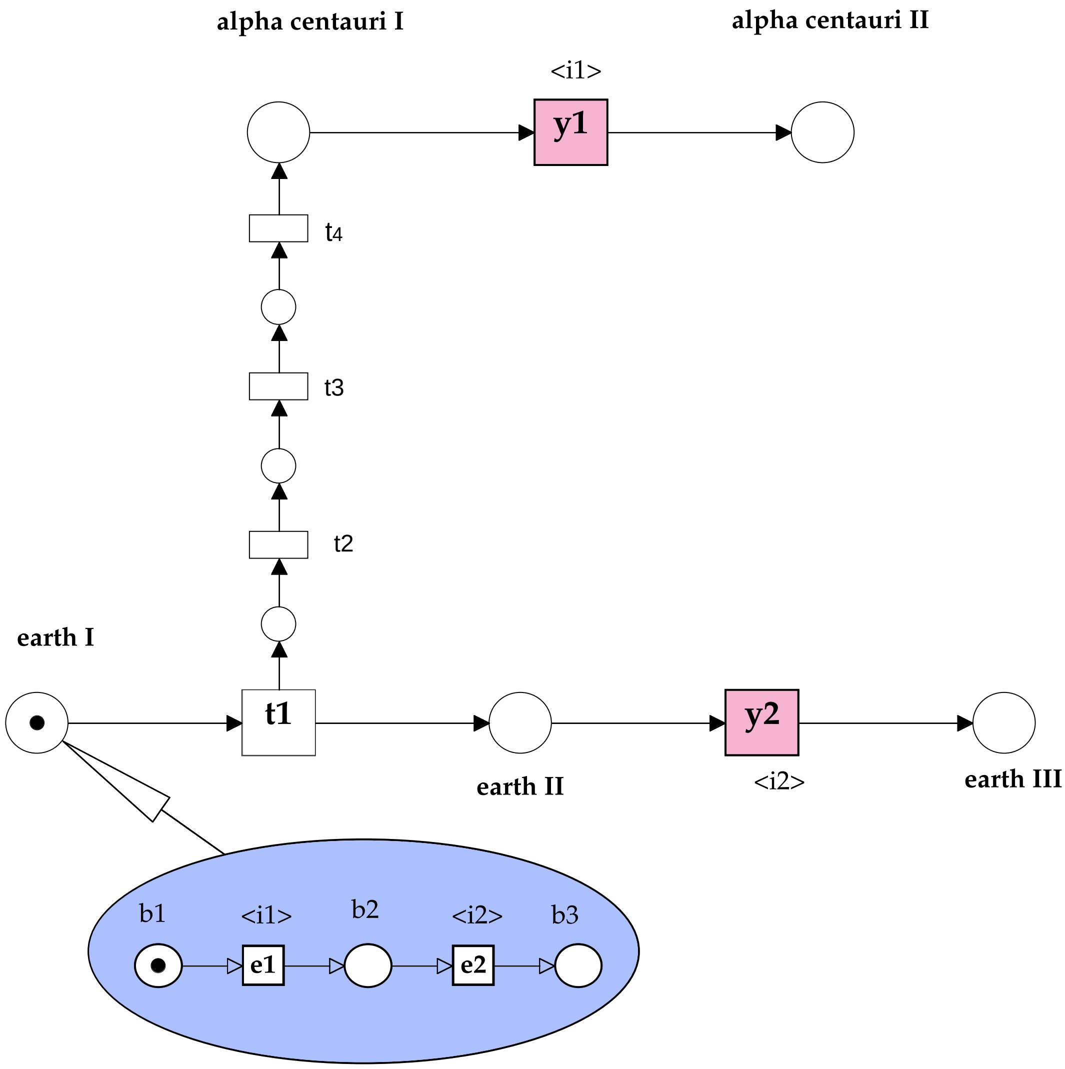

Figure 3: Elementary object system *alpha centauri*

$FS_{ref} := FS_{ref}(EOS) := \{\widetilde{w} \in \Gamma^* \mid \exists (\mathbf{M}', \mathbf{m}').\ (\mathbf{M}_0, \mathbf{m}_0) \xrightarrow{\widetilde{w}}_{ref} (\mathbf{M}', \mathbf{m}')\}$ *denotes the set of* occurrence sequences (firing sequences) *of EOS with respect to the reference semantics. We will use the projections* $pr_i(\widetilde{w})(1 \leq i \leq 2)$ *of* $\widetilde{w}$ *to its components, i.e. the maps* $pr_1(\widetilde{w}) \in T^*$ *and* $pr_2(\widetilde{w}) \in E^*$ *induced by the homomorphisms* $[t,x] \mapsto t, [\lambda, x] \mapsto \lambda$ *and* $[x,e] \mapsto e, [x, \lambda] \mapsto \lambda$*, respectively. We also use the notations* $\widetilde{w}_T := pr_1(\widetilde{w})$ *and* $\widetilde{w}_E := pr_2(\widetilde{w})$*.*

For the EOS of Figure 2 the following occurrence sequence is obtainable:

$$[\lambda, e_1], [t_1, \lambda], [t_3, e_3], [t_2, e_2], [t_4, e_4], [\lambda, e_5], [t_5, \lambda] \in FS_{ref}(con - task).$$

After this sequence, the initial bi-marking is reached again.

2.2 Value Semantics

Reference semantics and bi-markings, however, do not adequately reflect the nature of distributed computing. Consider for instance The EOS *alpha centauri* in Figure 3. Transition t_1 is interpreted as the beginning of a mission from earth to the star Alpha Centauri having an instance of the object system ON on board. After the occurrence sequence $[t_1, \lambda], [t_2, \lambda], [t_3, \lambda], [t_4, \lambda]$ the "task" e_1 is activated in interaction with the alpha-centauri-event y_1. After the occurrence of $[y_1, e_1]$ the bi-marking $(\mathbf{M}_1, \mathbf{m}_1)$ with $\mathbf{M}_1 = \{earth\,II, alpha\,centauri\,II\}$ and $\mathbf{m}_1 = \{b_2\}$ is reached, as shown in Figure 4a. Next $[y_2, e_2]$ is activated, which is quite strange for an observer on earth as there is no (modeled) communication from Alpha Centauri to earth.

Having such distributed applications in mind value semantics is more adequate. With the occurrence of transition t_1 two copies are generated in the output places which eventually reach the position as shown in Figure 4b. The last step before reaching the shown marking was the occurrence of $[y_1, e_1]$ which had an effect only on the copy of the object systems on Alpha Centauri. In consistency with our intuition transition y_2 on earth is not activated. It has been shown in [Val98] that a distributed analogy to bi-markings is not adequate here. Moreover, the marking of the object system is substituted by the process leading to this marking. Such markings are called *process-marking* or *p-marking*.

To give an example, in Figure 5 a p-marking is given for the EOS *con-task*, corresponding to the marking reached after the occurrence sequence $[\lambda, e_1], [t_1, \lambda], [t_3, e_3], [t_2, e_2]$. It shows the (partial) processes of concurrent task execution in the input places of transition t_4. (For a definition of processes we refer to the appendix.) Different to bi-markings, the history of the partial execution

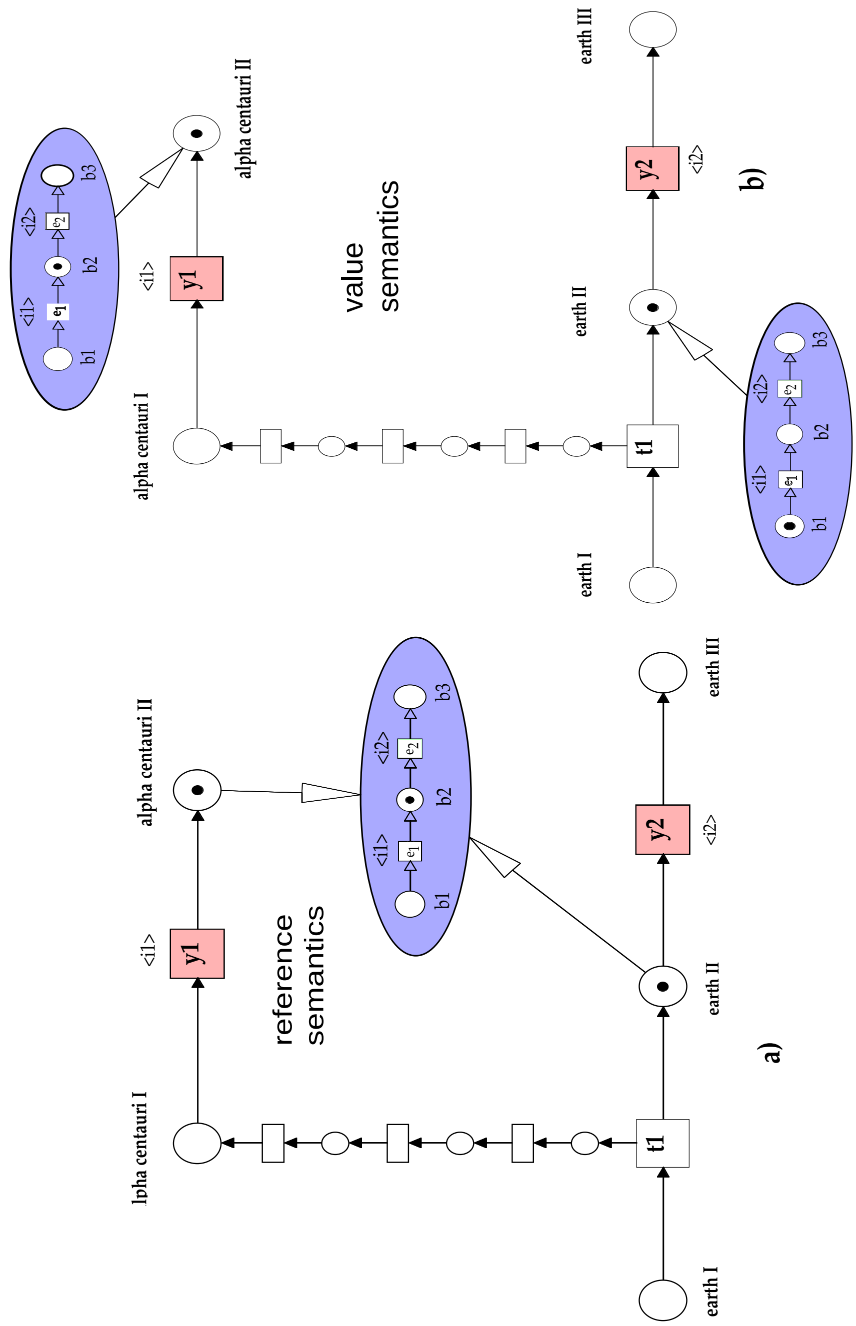

Figure 4: Elementary object system *alpha centauri* with reference and value semantics

is recorded, which allows for a more adequate detection of "fork/join-structures". (The small white circles in the places b_1 and p_1 indicate the initial marking.)

As in the case of a bi-marking we distinguish three cases: a) *interaction*, b) *transport* (or system autonomous transition) and c) *autonomous* (or object autonomous) *transition*. An interaction $[t, e]$ is activated if all input places in $^\bullet t$ contain a process of the object net ON such that the input processes can be composed to a consistent process. This is formalized by the *least upper bound* "*lub*" of these processes. The *lub* of a set of processes is the smallest process which is a continuation of all these processes. The corresponding relation, denoted by $\preceq$, is formally introduced in the appendix: for processes $proc_1, proc_2 \in PROC(ON)$ the holding of $proc_1 \preceq proc_2$ denotes that $proc_1$ is an initial subprocess of $proc_2$. The lub-operation is also introduced in the appendix. $lub(proc_1, proc_2)$ does not necessarily exist, but if it exists it is a "consistent" composition of both processes, which can be interpreted as the "join" with respect to a "fork" in the past.

Definition 2.4 *A* process-marking (p-marking) *of a unary elementary object system* $EOS = (SN, ON, \rho)$, *where* $SN = (P, T, W, \mathbf{M}_0)$ *and* $ON = (B, E, F, \mathbf{m}_0)$, *is a partial mapping* $\mu : P \hookrightarrow PROC(ON)$, *giving to each place* $p \in dom\ \mu$ *of the system net a process* $\mu(p)$ *of the object net.* $dom\ \mu \subseteq P$ *is the* associated system net marking. *For the initial p-marking* μ_0 *we assume dom* $\mu_0 = \mathbf{M}_0$ *and* $\mu_0(p) = proc_{\mathbf{m}_0}$ *for each* $p \in dom\ \mu$, *where marking is the initial process corresponding to* $\mathbf{m}_0$ *(for the definition of dom and* $proc_{\mathbf{m}_0}$ *see the appendix).*

We will use the following definition of the set of input- and output-processes *of a given transition t:* $^\oplus t := \{\mu(p) \mid p \in {^\bullet t} \cap dom\ \mu\}$ *and* $t^\oplus := \{\mu(p) \mid p \in t^\bullet \cap dom\ \mu\}$.

Definition 2.5 *Given an unary elementary object system EOS as in Definition 2.4, a system net transition t, an object net transition e and a p-marking* μ. *To define the successor marking relations* $\mu \xrightarrow{[t,e]} \mu'$, $\mu \xrightarrow{[t,\lambda]} \mu'$ *and* $\mu \xrightarrow{[\lambda,e]} \mu'$ *we proceed in three steps:*

a) Interaction*:* $t \in T$, $e \in E$, $(t, e) \in \rho$

1. $[t, e]$ *is* activated *in* μ $(\mu \xrightarrow{[t,e]})$, *if* $^\bullet t \subseteq dom\ \mu$, $t^\bullet \cap dom\ \mu = \emptyset$ *hold and both* $\sqcup {^\oplus t}$ *and* $(\sqcup {^\oplus t}) \circ e$ *exist. (i.e. all input places of t contain processes and the output places are empty. Their lub exists and activates e. For the definition of the lub-operation* $\sqcup$ *see the appendix.)*
2. $[t, e]$ occurs *and transforms* μ *into the successor p-marking* μ'*:* $(\mu \xrightarrow{[t,e]} \mu')$, *if* $[t, e]$ *is activated in* μ *and* μ' *is defined by dom* $\mu' :=$

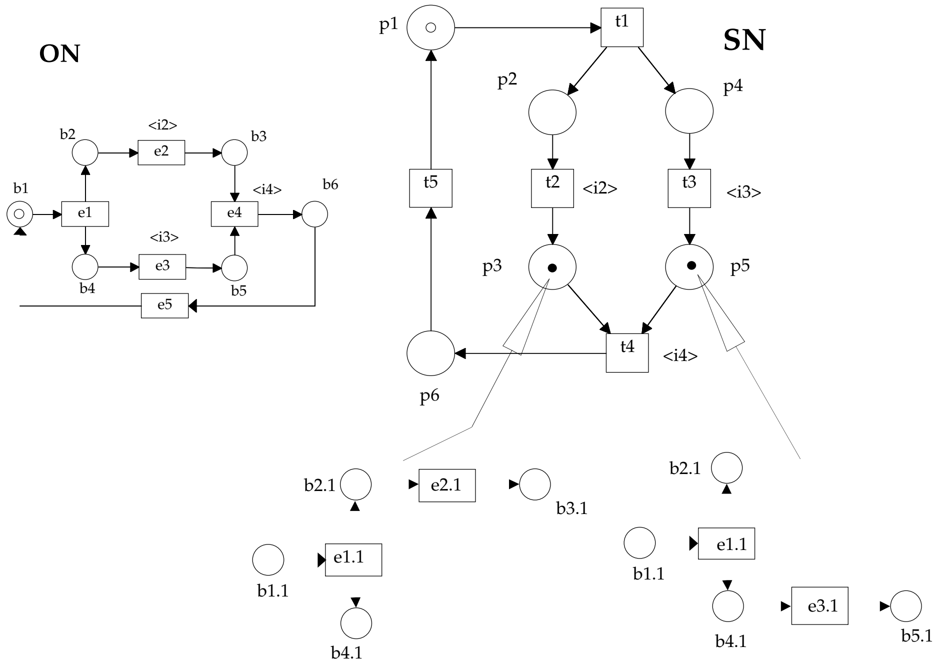

Figure 5: Elementary object system *con-task* with P-marking

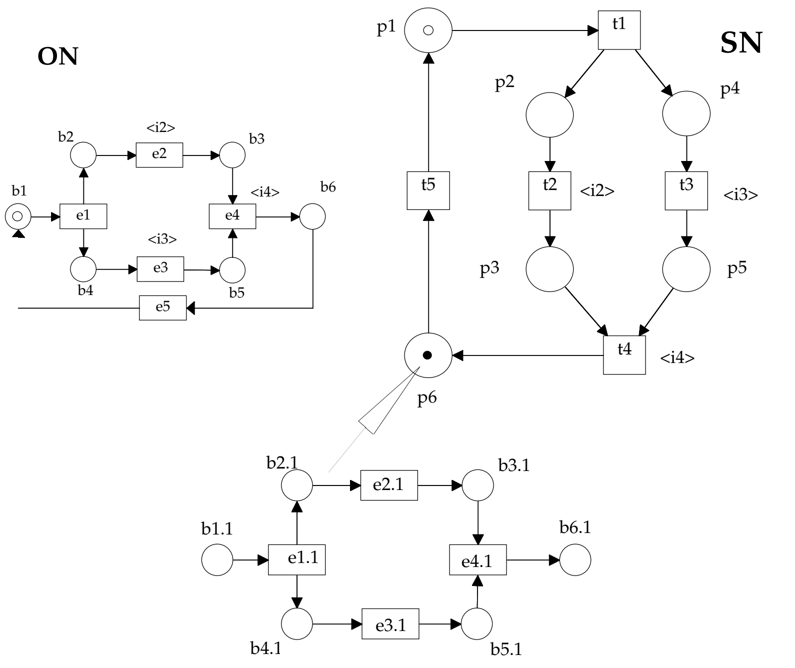

Figure 6: Successor p-marking of Figure 5

$(dom\ \mu \backslash {}^{\bullet}t) \cup t^{\bullet}$ *and for* $p \in dom\ \mu'$ *let be*

$$\mu'(p) := \begin{cases} (\sqcup {}^{\oplus}t) \circ e & for\ p \in t^{\bullet} \\ \mu(p) & otherwise \end{cases}$$

b) Transport*:* $t \in T$, $t\rho = \emptyset$

See part a) 1. and 2. and substitute λ *for* e*. Then* $\sqcup({}^{\oplus}t) = \sqcup({}^{\oplus}t) \circ \lambda$ *and the corresponding conditions coincide. For the definition of* $proc \circ \lambda$ *see the appendix. (A transport is like an interaction but without the prolongation of the object net process by e.)*

c) Object-autonomous event*:* $e \in E$, $\rho e = \emptyset$

1. $[\lambda, e]$ *is activated in* μ $(\mu \xrightarrow{[\lambda,e]})$ *, if for some* $p \in dom\ \mu$ *the process* $\mu(p) \circ e$ *exists. (e is activated in* $\mu(p)$*.)*
2. $[\lambda, e]$ occurs *and transforms* μ *into the successor p-marking* μ'*:* $(\mu \xrightarrow{[\lambda,e]} \mu')$*, if* $[\lambda, e]$ *is activated in* μ *and* μ' *is defined by* $dom\,\mu' := dom\,\mu$ *and for* $p_1 \in dom\ \mu'$ *let be* $\mu'(p_1) := \begin{cases} \mu(p) \circ e & for\ p = p_1 \\ \mu(p_1) & otherwise \end{cases}$.
   (A single process in a place p is enlarged by e.)

According to case a) of Definition 2.5 the pair $[t_4, e_4]$ is activated in the p-marking of Figure 5. The successor p-marking is given in Figure 6. With the slight modification of the object net EOS *con-task-mod* in Figure 7 a p-marking is given that is *not* activated for $[t_4, e_4]$ as the processes in the input places of t_4 have no *lub*.

Definition 2.6 *The successor p-marking relation* $\mu \xrightarrow{[\alpha,\beta]} \mu'$ *is inductively extended to finite sequences* $\widetilde{w} \in \Gamma^*$ *(where* $\Gamma := (T \cup \{\lambda\}) \times (E \cup \{\lambda\})) \backslash [\lambda, \lambda]$ *and* $[\lambda, \lambda]$ *denotes the neutral element of the free monoid* Γ^**):*

- $\mu \xrightarrow{[\alpha,\beta]}_{val} \mu$ *if* $[\alpha, \beta] = [\lambda, \lambda]$ *and*
- $\mu \xrightarrow{\widetilde{w}[\alpha,\beta]}_{val} \mu'$ *if* $\exists \mu''.\ \mu \xrightarrow{\widetilde{w}}_{val} \mu'' \wedge \mu'' \xrightarrow{[\alpha,\beta]} \mu'$ *for* $\widetilde{w} \in \Gamma^*$ *and* $[\alpha, \beta] \in \Gamma$

$FS_{val} := FS_{val}(EOS) := \{\widetilde{w} \in \Gamma^* \mid \exists \mu'.\ \mu_0 \xrightarrow{\widetilde{w}}_{val} \mu'\}$ *denotes the set of* occurrence sequences (firing sequences) *of EOS with respect to the value semantics.*

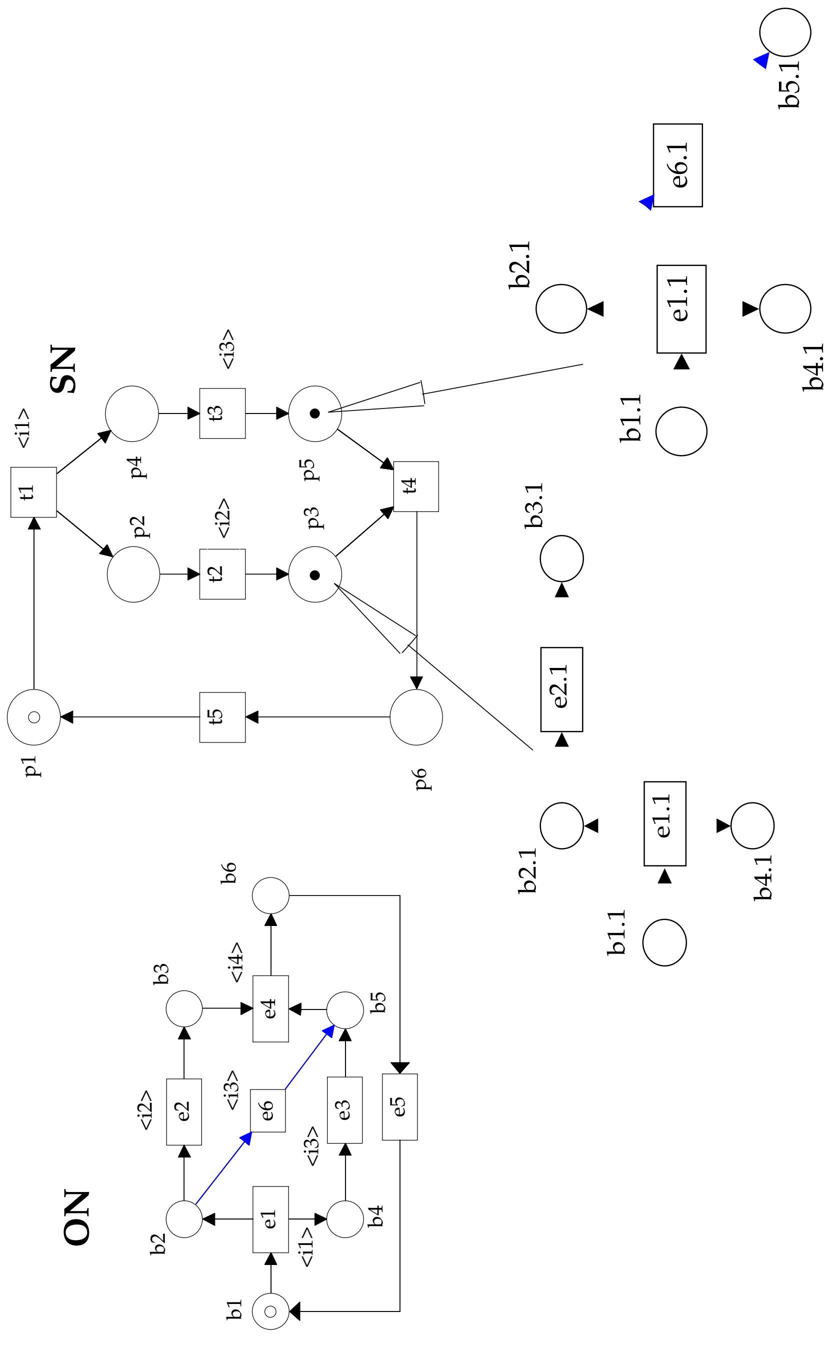

Figure 7: Elementary object system *con-task-mod* with p-marking

3 Processes of Elementary Object Systems

For the rest of the paper, we will use the following notations whenever possible. For an elementary object system $EOS = (SN, ON, \rho)$

- a process $proc_{SN} \in PROC(SN)$ of the system net $SN = (P, T, W, \mathbf{M}_0)$ is denoted by $proc_{SN} = (X_P, Y_T, Z_W, \phi_{SN})$ with $\phi_{SN} : X_P \cup Y_T \to P \cup T$. $<_{proc_{SN}} := Z_W^+$ is the causal ordering.
- a process $proc_{ON} \in PROC(ON)$ of the object net $ON = (B, E, F, \mathbf{m}_0)$ is denoted by $proc_{ON} = (X_B, Y_E, Z_F, \phi_{ON})$ with $\phi_{ON} : X_B \cup Y_E \to B \cup E$. $<_{proc_{ON}} := Z_F^+$ is the causal ordering.
- $\rho \subseteq T \times E$ is the *interaction relation.*

Definition 3.1 *A triple* $\Theta := (proc_{SN}, proc_{ON}, \varphi)$ *is called a* process pair *of EOS if* $proc_{SN} \in PROC(SN)$, $proc_{ON} \in PROC(ON)$ *and* $\varphi : Y_T \hookrightarrow Y_E$ *is a partial mapping with* $dom(\varphi) = \phi_{SN}^{-1}(dom\ \rho), range(\varphi) = \phi_{ON}^{-1}(range\ \rho)$ *and* $(\phi_{SN}(y), \phi_{ON}(\varphi(y))) \in \rho$ *for all* $y \in dom\ \varphi$.

Definition 3.2 *Given a unary elementary object system EOS (as above) and an occurrence sequence* $\widetilde{w} = [\alpha_1, \beta_1][\alpha_2, \beta_2] \dots [\alpha_{n-1}, \beta_{n-1}] \in FS_{ref}(EOS)$ *leading to the bi-marking* $(\mathbf{M}, \mathbf{m})$ *w.r.t. the reference semantics, we inductively define a* ref-process $proc_{ref}(\widetilde{w}) = (proc_{SN}, proc_{ON}, \varphi)$, *where* $proc_{SN}$ *and* $proc_{ON}$ *are processes of* SN *and* ON, *respectively, and* $\varphi : Y_T \hookrightarrow Y_E$.

a) If $n = 1$ *(i.e.* $\widetilde{w} = [\lambda, \lambda]$*) then* $proc_{ref}(\widetilde{w}) = (proc_{\mathbf{M}_0}, proc_{\mathbf{m}_0}, \emptyset)$

b) If $n > 1$ *and* $proc_{ref}(\widetilde{w}) = (proc_1, proc_2, \varphi)$ *then for* $(\mathbf{M}, \mathbf{m}) \xrightarrow{[\alpha_n, \beta_n]}$ *we define* $proc_{ref}(\widetilde{w}[\alpha_n, \beta_n]) := (proc_1 \circ \alpha_n, proc_2 \circ \beta_n, \varphi \cup A)$ *with*

$$A := \begin{cases} \{(\chi(proc_1, t), \chi(proc_2, e))\} & \textit{for } [\alpha_n, \beta_n] = [t, e] \\ \emptyset & \textit{otherwise} \end{cases}$$

Remark For the definition of χ see the appendix ($\chi(proc, t)$ denotes the name of the transition that enlarges the process *proc*, when the occurrence of t is represented, i.e. $\phi(\chi(proc, t)) = t$). Note that the ref-process $proc_{ref}(\widetilde{w})$ of $\widetilde{w}$ is uniquely defined and φ is injective. Figure 8 shows a ref-process for the *EOS con-task.*

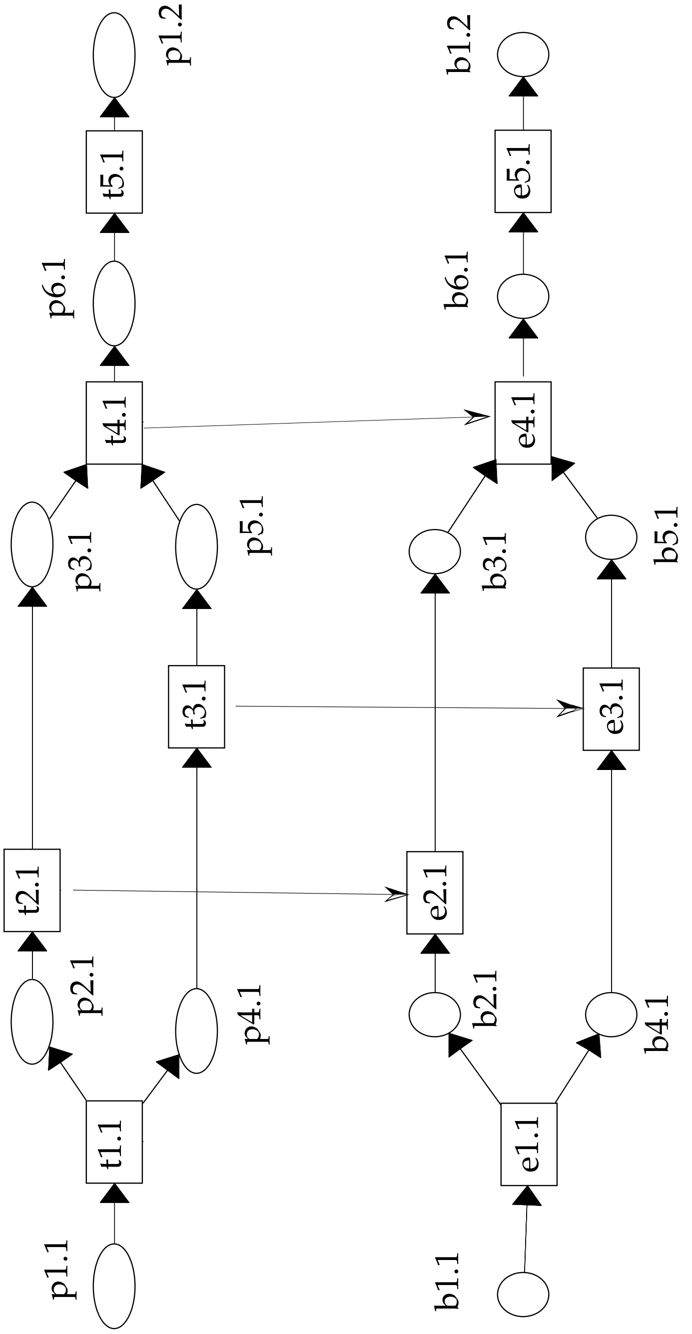

Figure 8: A process of *con-task* in reference semantics

Definition 3.3 *Given a unary elementary object system EOS with initial p-marking and an occurrence sequence* $\widetilde{w} = [\alpha_1, \beta_1][\alpha_2, \beta_2] \ldots [\alpha_{n-1}, \beta_{n-1}] \in FS_{val}(EOS)$ *leading to the p-marking* μ *w.r.t. the value semantics, we inductively define a set of* val-processes $proc_{val}(\widetilde{w}) = (proc_{SN}, \eta)$*, where* $proc_{SN} = (X_P, Y_T, Z_W, \phi_{SN})$ *is a process of SN and* $\eta : X_P \to 2^{PROC(ON)}$ *is a mapping.*

a) If $n = 1$ *(i.e.* $\widetilde{w} = [\lambda, \lambda]$*) then* $proc_{val}(\widetilde{w}) = (proc_{\mathbf{M_0}}, \eta_0)$ *where* $\eta_0(x) := proc_{\mathbf{m_0}}$ *for all* $x \in Min(proc_{\mathbf{M_0}})$

b) If $n > 1$ *and* $proc_{val}(\widetilde{w}) = (proc_1, \eta)$*, then for* $\mu \xrightarrow{[t,\beta_n]}$ *with* $\beta_n \in E \cup \{\lambda\}$ *we define* $proc_{val}(\widetilde{w}[t, \beta_n]) = (proc_1 \circ t, \eta')$ *with* $\eta'(x) = \eta(x)$ *for places in* $proc_1$*. To define* $\eta'(x)$ *for the new places of* $proc_1 \circ t$*, let* $t.n := \chi(proc_1, t)$*. Then* $\eta'(x_1) := (\bigsqcup_{x \in {}^\bullet t} \eta(x)) \circ \beta_n$ *for* $x_1 \in (t.n)^\bullet$*.*

c) If $n > 1$ *and* $proc_{val}(\widetilde{w}) = (proc_1, \eta)$*, then for* $\mu \xrightarrow{[\lambda,e]}$ *a transition e is activated in some process* $proc_{ON} \in \mu(p)$*.*
We define $proc_{val}(\widetilde{w}[\lambda, e]) := (proc_1, \eta')$ *with*

$$\eta'(x) := \begin{cases} \eta(x) \cup \{proc_{ON} \circ e\} & \text{when } x \in Max(proc_1) \cap \phi_{SN}^{-1}(p) \\ \eta(x) & \text{otherwise} \end{cases}$$

for $x \in X_P$ *of* $proc_1$*.*

Remark Note that due to the choice of the place p in step c) in Definition 3.3 the val-process $proc_{val}(\widetilde{w})$ of $\widetilde{w}$ is *not* uniquely defined and defines a set of val-processes. Figure 9 shows a val-process for the *EOS con-task*. $\eta(p_1.1)$ and $\eta(p_1.2)$ are sets with more than one element due to the autonomous transitions $e_1.1$ and $e_5.1$, respectively.

Definition 3.4 *Let* $proc_{val}(\widetilde{w}) = (proc_{SN}, \eta)$ *be the val-process of a unary EOS, as introduced in Definition 3.3, such that the lub* $proc_\omega := \sqcup\{\eta(x) \mid x \in X_P\}$ *exists. Then the triple* $proc_{rv}(\widetilde{w}) = (proc_{SN}, proc_{ON}, \varphi)$ *is called a* rv-process-representation, *if* $proc_{ON} = proc_\omega$ *and* φ *is defined as follows. Define* $\varphi := \emptyset$ *initially and construct* φ *with the inductive creation of* $proc_\omega$ *in Definition 3.3 by adding the pair* (y, y') *in each step when applying case b) with* $\beta_n = e \in E$*, y as defined there and* $y' := \chi(\bigsqcup_{x \in {}^\bullet t} \eta(x)), e)$*.*

Figure 8 shows a rv-process-representation of the val-process in Figure 9, which is also a ref-process in this particular case.

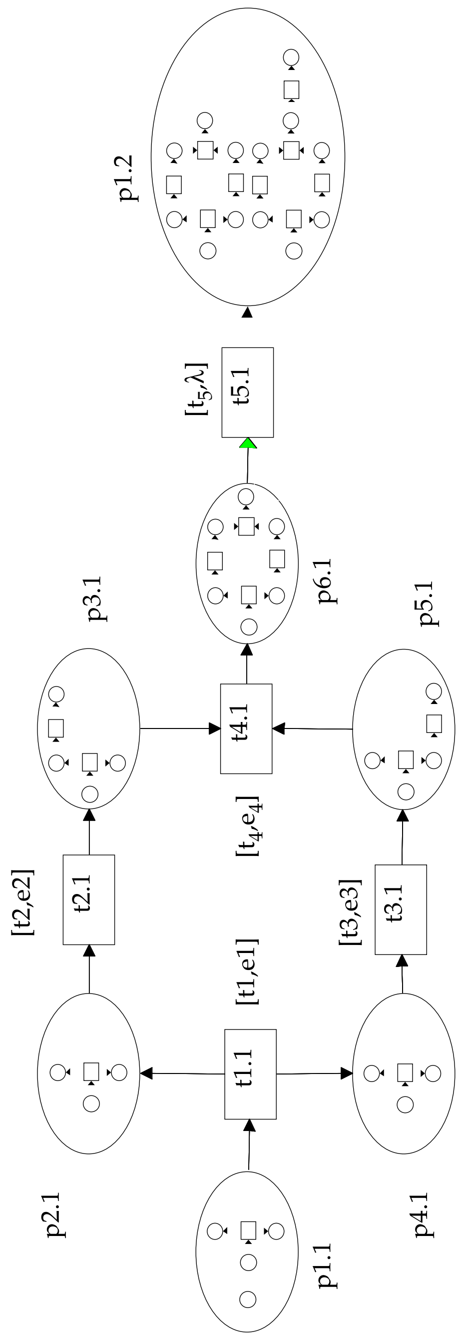

Figure 9: A process of *con-task* in value semantics

4 Characterizing ref-processes

In [Val98] a sufficient condition was given for a triple $\Theta := (proc_{SN}, proc_{ON}, \varphi)$ (Definition 3.1) to be a val-process. Here we discuss the same question for ref-processes.

Definition 4.1 *Given a triple $\Theta := (proc_{SN}, proc_{ON}, \varphi)$ as defined in the beginning of section 3. The following relation* strongly before *(denoted: $\ll$) is defined on the transition set Y_E of $proc_{SN}$:*

$$\begin{aligned} \forall y_1, y_2 \in Y_T . y_1 \ll y_2 \Leftrightarrow: \exists k > 1.\ \exists \bar{y}_1, \bar{y}_2, \ldots \bar{y}_k \in Y_T . y_1 \leq_{proc_{SN}} \bar{y}_1 \wedge \\ \varphi(\bar{y}_1) <_{proc_{ON}} \varphi(\bar{y}_2) \wedge \bar{y}_2 \leq_{proc_{SN}} \bar{y}_3 \wedge \\ \varphi(\bar{y}_3) <_{proc_{ON}} \varphi(\bar{y}_4) \wedge \bar{y}_4 \leq_{proc_{SN}} \bar{y}_5 \wedge \\ \ldots \\ \varphi(\bar{y}_{k-3}) <_{proc_{ON}} \varphi(\bar{y}_{k-2}) \wedge \bar{y}_{k-2} \leq_{proc_{SN}} \bar{y}_{k-1} \wedge \\ \varphi(\bar{y}_{k-1}) <_{proc_{ON}} \varphi(\bar{y}_k) \wedge \bar{y}_k \leq_{proc_{SN}} y_2 \end{aligned}$$

Theorem 4.2 *: Let $\Theta := (proc_{SN}, proc_{ON}, \varphi)$ be a process pair of an EOS (Def.3.1) such that the partial mapping $\varphi : Y_T \hookrightarrow Y_E$ is injective. Then Θ is a ref-process (Def.3.2) iff the following* "reference semantics consistency condition" *(RSCC) holds:*

$$\forall y_1, y_2 \in dom(\varphi).\ \varphi(y_1) <_{proc_{ON}} \varphi(y_2) \Rightarrow \neg(y_2 \ll y_1)$$

Proof

Let be $\Theta := (proc_{SN}, proc_{ON}, \varphi)$ process pair of an EOS (Def.3.1) such that $proc_{SN} \in PROC(SN)$, $proc_{ON} \in PROC(ON)$ and the partial mapping $\varphi : Y_T \hookrightarrow Y_E$ is injective. Assuming the property $RCSS$ of Theorem 4.2 we have to show that Θ is a ref-Process (Definition 3.2). We therefore inductively construct an occurrence sequence $\widetilde{w} = [\alpha_1, \beta_1][\alpha_2, \beta_2] \ldots [\alpha_n, \beta_n] \in FS_{ref}(EOS)$ leading to the bi-marking $(\mathbf{M}, \mathbf{m})$ w.r.t. the reference semantics such that $\Theta = proc_{ref}(\widetilde{w})$.

For $k = 0$ we obviously have $\widetilde{w}_k = [\lambda, \lambda]$.

For $0 \leq k < n$ we assume $\Theta^k := (proc^k_{SN}, proc^k_{ON}, \varphi^k)$ with $proc^k_{SN} \preceq proc_{SN}$, $proc^k_{ON} \preceq proc_{ON}$ and φ^k is a restriction of φ. By induction hypothesis there is a sequence $\widetilde{w}_k = [\alpha_1, \beta_1][\alpha_2, \beta_2] \ldots [\alpha_k, \beta_k] \in FS_{ref}(EOS)$ leading to the bi-marking $(\mathbf{M}_k, \mathbf{m}_k)$ such that $\Theta_k = proc_{ref}(\widetilde{w}_k)$.

We have to construct a prolongation $[\alpha_{k+1}, \beta_{k+1}]$ of $\widetilde{w}_k$ such that

1) $proc^k_{SN} \circ y \preceq proc_{SN}$ with $y \in \{\phi_{SN}(\alpha_{k+1}), \lambda\}$

2) $proc^k_{ON} \circ v \preceq proc_{ON}$ with $v \in \{\phi_{ON}(\beta_{k+1}), \lambda\}$ and

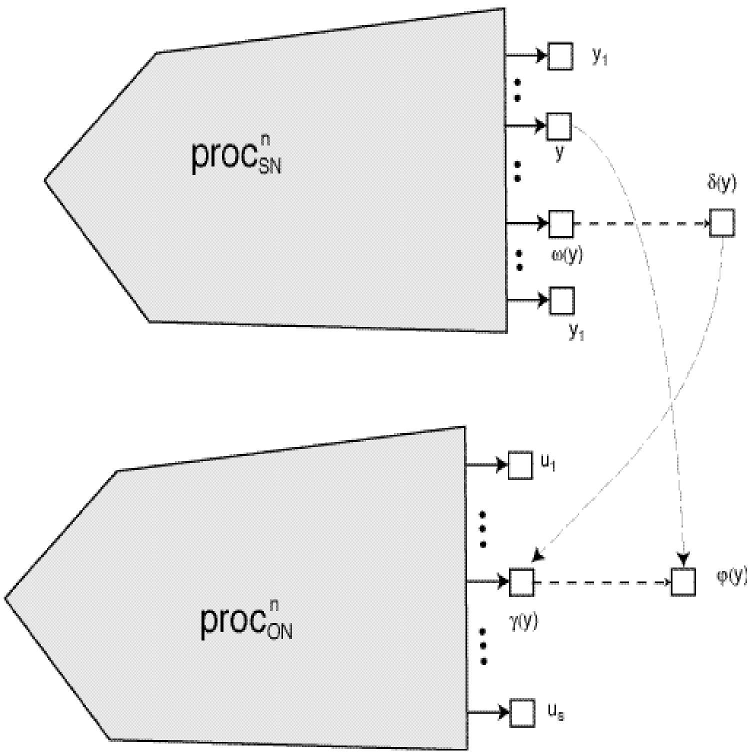

Figure 10: Symbolic representation of subcase 2.2 of the proof of Theorem 4.2

3) φ^k is a restriction of φ.

case 1: There is a prologation by an autonomous transition, i.e. $y \notin dom\ \phi$ with $proc^k_{SN} \circ y \preceq proc_{SN}$ or $v \notin range\ \phi$ with $proc^k_{ON} m \circ v \preceq proc_{ON}$. Then we choose the appropriate prologation of $\widetilde{w}$ by $[\alpha_{k+1}, \beta_{k+1}] = [t, \lambda]$ or $[\alpha_k, \beta_k] = [\lambda, e]$.

case 2: not case 1:

subcase 2.1: There are $t \in T$ and $e \in E$ satisfying $(t, e) \in \rho$ with $proc^k_{SN} \circ y \preceq proc_{SN}$ for $\phi_{SN}(y) = t$ and $proc^k_{ON} \circ v \preceq proc_{SN}$ for $\phi_{ON}(v) = e$. Then we choose $[\alpha_{k+1}, \beta_{k+1}] = [t, e]$.

subcase 2.2: There is no such prolongation of the processes as in the cases 1 and 2.1. We will prove that this subcase is in contradiction with the assumed property $RCSS$.

Define $\widetilde{Y} := \{y \mid proc^k_{SN} \circ y \preceq proc_{SN}\}$ as the set of transitions activated in $proc^k_{SN}$ and similarly $\widetilde{U} := \{u \mid proc^k_{ON} \circ u \preceq proc_{ON}\}$ as the set of transitions activated in $proc^k_{ON}$. Both sets are nonempty as $k < n$ and $\varphi : Y_T \hookrightarrow Y_E$ is injective. For each $y \in \widetilde{Y}$ we have $\varphi(y) \notin \widetilde{U}$ as we would be in case 2.1 otherwise. Furthermore $\varphi(y)$ cannot lie in $proc^k_{ON}$. Hence there is some $u \in \widetilde{U}$ with $u \leq_{ON} \varphi(y)$. For each $y \in \widetilde{Y}$ the choice of such a transition u is denoted by $\gamma(y)$. (See the symbolic representation of these constructions in Figure 10.) Since $\gamma(y) \in range(\varphi)$ and φ is injective, we can define $\delta(y) := \varphi^{-1}(\gamma(y)) \notin \widetilde{Y}$, where $\delta(y) \notin \widetilde{Y}$ holds as case 2.1 is excluded here. By a similar argument as before we can find an element $\omega(y) \in \widetilde{Y}$ such that $\omega(y) \leq_{SN} \delta(y)$. We thus have constructed a mapping $\omega : \widetilde{Y} \rightarrow \widetilde{Y}$ which certainly contains a permutation

$$\begin{pmatrix} y_1 & y_2 & \dots & y_r \\ \omega(y_1) & \omega(y_2) & \dots & \omega(y_r) \end{pmatrix}$$

on a subset of $\widetilde{Y}$ i.e. there is a fixpoint $\hat{y} \in \widetilde{Y}$ satisfying $\omega^n(\hat{y}) = \hat{y}$ for some $n \geq 1$.

We have $\varphi(\delta(\hat{y})) \leq_{ON} \varphi(\hat{y})$ and we will prove next that $\hat{y} \ll \varphi(\hat{y})$, which is in contradiction to the assumption of the property $RSCC$. Therefore case 2.2 is impossible.

To show $\hat{y} \ll \delta(\hat{y})$ let be $\widetilde{y} := \omega(\hat{y}) = \omega^{n-1}(\hat{y})$. Furthermore we define:

$$\begin{array}{ll} \bar{y}_1 := \delta(\widetilde{y}) = \delta(\omega^{n-1}(\hat{y})) & \bar{\bar{y}}_1 := \omega^{n-1}(\hat{y}) \\ \bar{y}_2 := \delta(\omega^{n-1}(\hat{y})) & \bar{\bar{y}}_2 := \omega^{n-2}(\hat{y}) \\ \dots & \dots \\ \bar{y}_{n-1} := \delta(\omega^{n-(n-1)}(\hat{y})) & \bar{\bar{y}}_{n-1} := \omega(\hat{y}) \\ \bar{y}_n := \delta(\hat{y}) & \end{array}$$

This finally gives:

$$\begin{array}{lll} \hat{y} \leq_{SN} \bar{y}_1 = \delta(\hat{y}) & \wedge & \varphi(\bar{y}_1) \leq_{ON} \varphi(\bar{\bar{y}}_1) \wedge \bar{\bar{y}}_1 \leq_{SN} \bar{y}_2 \\ & \wedge & \varphi(\bar{y}_2) \leq_{ON} \varphi(\bar{\bar{y}}_2) \wedge \bar{\bar{y}}_2 \leq_{SN} \bar{y}_3 \\ & \dots & \\ & \wedge & \varphi(\bar{y}_{n-1}) \leq_{ON} \varphi(\bar{\bar{y}}_{n-1}) \wedge \bar{\bar{y}}_{n-1} \leq_{SN} \bar{y}_n = \delta(\hat{y}) \end{array}$$

$\diamondsuit$

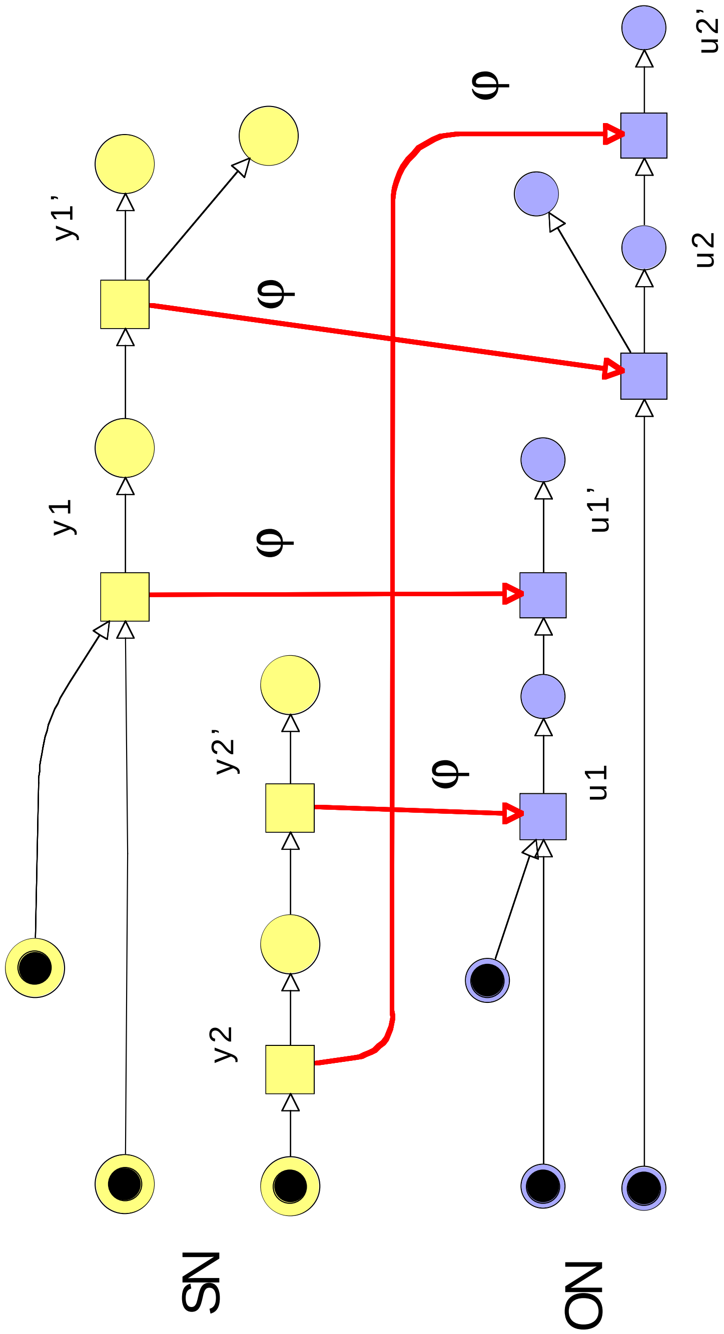

Figure 11: Counter example to Theorem 4.2

In Figure 11 a counter example is given. The system net SN as well as the object net ON are causal nets and they are isomorphic to their unique maximal processes. But with the represented mapping φ the process pair $\Theta := (proc_{SN}, proc_{ON}, \varphi)$ is not a ref-process since the condition RSCC is violated: $\varphi(y_2') <_{proc_{ON}} \varphi(y_1)$ but $y_1 <_{proc_{SN}} y_1' \wedge \varphi(y_1') <_{proc_{ON}} y_2 \wedge y_2 <_{proc_{SN}} y_2'$ hence $y_1 \ll y_2'$.

5 From Reference Semantics to Value Semantics

In this section we give a sufficient condition for an occurrence sequence with respect to the reference semantics to be also an occurrence sequence with respect to the value semantics.

Definition 5.1 *Given a triple $\Theta := (proc_{SN}, proc_{ON}, \varphi)$ (Def. 3.1), such that φ is injective. Then a (total) mapping $\psi : Y_E \to X_P \cup Y_T$ is said to have the* full morphism property (FMP) *if*

a) $\forall e \in Y_E.\ e \in range\, \varphi \Rightarrow \psi(e) \in Y_T \wedge \psi(e) = \varphi^{-1}(e)$

b) $\forall e \in Y_E.\ e \notin range\, \varphi \Rightarrow \psi(e) \in X_P$

c) $\forall e_1, e_2 \in Y_E.\ e_1 \leq_{ON} e_2 \Rightarrow \psi(e_1) \leq_{SN} \psi(e_2)$

A partial mapping $\psi_0 : Y_E \hookrightarrow X_P \cup Y_T$ is said to be fully extensible *iff there is an extension to a full morphism.*

The existence of a morphism as given by the full morphism property is characteristic for an occurrence sequence w.r.t. the value semantics. To give an example consider the *EOS alpha centauri* under reference semantics in Fig. 4a). As the mapping ψ of Def. 5.1 has to extend φ^{-1} with $\varphi^{-1}(e_1) = y_1, \varphi^{-1}(e_2) = y_2$ it cannot be a morphism.

Theorem 5.2 *Let EOS be a unary elementary object system (as before) and let $\widetilde{w} \in FS_{ref}(EOS)$ be an occurrence sequence w.r.t. the reference semantics. If for the corresponding ref-process $proc_{ref}(\widetilde{w}) = (proc_{SN}, proc_{ON}, \varphi)$ (Def. 3.2) the partial mapping φ^{-1} is fully extensible (Def. 5.1), then $\widetilde{w}$ is also an occurrence sequence $\widetilde{w} \in FS_{val}(EOS)$ w.r.t. the value semantics and $proc_{ref}(\widetilde{w})$ is a val-process in rv-process-representation.*

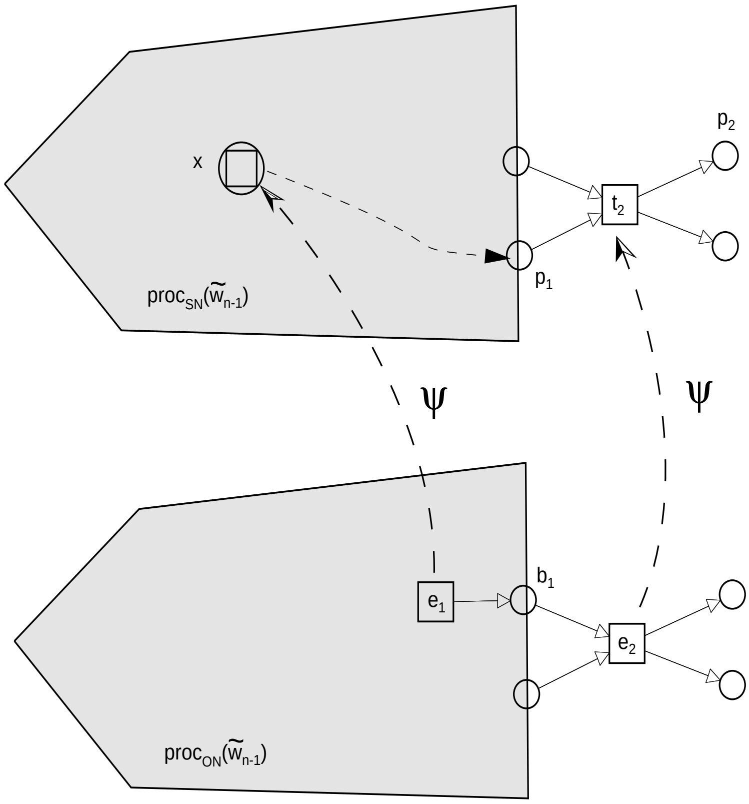

Figure 12: Symbolic representation of cases a) and b) of the proof of Theorem 5.2.

Proof
To prove the theorem from $\widetilde{w} \in FS_{ref}(EOS)$ and its ref-process $proc_{ref}(\widetilde{w}) = (proc_{SN}, proc_{ON}, \varphi)$ we will construct a val-process (Def. 3.3) $(proc', \eta)$, such that $proc_{val}(\widetilde{w}) = (proc', \eta)$. Define $proc' := proc_{SN}$ and $\eta : X_P \to 2^{PROC(ON)}$ as follows.

First we have to introduce some notation w.r.t. the process $proc_{ON} = (X_B, Y_E, Z_F, \phi_{ON})$ (which applies to all processes as a general definition, however). The *open closure* of a set $X \subseteq X_B \cup Y_E$ is the set $ocl(X) := \bigcup\{{}^\bullet x \mid x \in X \cap Y_E\} \cup \bigcup\{x^\bullet \mid x \in X \cap Y_E\} \cup X$. Intuitively the open closure is obtained by attaching all input and output places to the transitions of X. Using this notation, for a set $A \subseteq Y_E$ of transitions of the process $proc_{ON}$ we define the net $cl(proc_{ON}, A)$ by attaching all input and output places, i.e.:

$cl(proc_{ON}, A) := (X_1, Y_1, Z_1, \phi_1)$ with :

- $X_1 := ocl(A) \cap X_B$
- $Y_1 := A$
- $Z_1 := Z_F \cap ((X_1 \times A) \cup (A \times X_1))$
- $\phi_1 := \phi_{ON|(X_1 \cup A)}$

When $proc_{ON}$ is obvious from the context we will write $cl(A)$ for $cl(proc_{ON}, A)$.

We now proceed in the definition of the map $\eta : X_P \to 2^{PROC(ON)}$. For each $x \in X_P$ the set $\eta(x)$ will contain at least one process of ON, which is called *input process of* x and is denoted by $proc_\alpha(x)$. It is defined by

$$proc_\alpha(x) := cl(A_\alpha(x))) \tag{1}$$

with

$$A_\alpha(x) := \{\psi^{-1}(y) \mid y \in Y_T \cup X_P \wedge y <_{SN} x\} \subseteq Y_E \tag{2}$$

Intuitively, the input process of x is the process that marks x at the first. Analogously, the *output process* will be the process that is added to x at the last. The input and output processes only differ in autonomous transitions. The output process is defined by enlarging $proc_\alpha(x)$ by autonomous transitions:

$$proc_\omega(x) := cl(A_\omega(x))) \tag{3}$$

with

$$A_\omega(x) := \{\psi^{-1}(y) \mid y \in Y_T \cup X_P \wedge y \leq_{SN} x\} \subseteq Y_E \tag{4}$$

Intuitively $\{e_1.n_1, e_2.n_2, \ldots, e_k.n_k\} := \psi^{-1}(x)$ is the set of autonomous transitions that occur when the process is in the place x. Therefore $proc_\alpha(x) = proc_\omega(x)$ if $k = 0$ i.e.

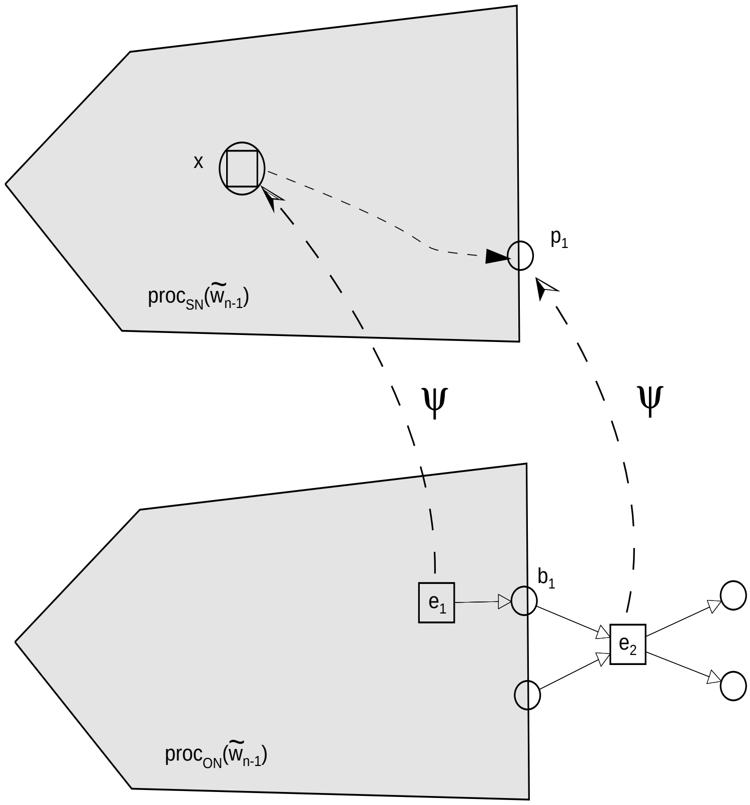

Figure 13: Illustration of case c) of the proof of Theorem 5.2.

$\psi^{-1}(x) = \emptyset$. This follows also formally from the equations (2) and (4). If $\psi^{-1}(x)) \neq \emptyset$ then the set $\{e_1.n_1, e_2.n_2, \ldots, e_k.n_k\}$ is totally ordered by the order $< := <_{\widetilde{w}_E}$ induced by the projection $pr_2(\widetilde{w}) = \widetilde{w}_E$, say $e_{i_1} < e_{i_2} < \ldots < e_{i_k}$ (see appendix for the induced order $<_w$ and the text after Def. 2.3 for the projection $pr_2(\widetilde{w}) = \widetilde{w}_E$). We will show that e_{i_1} is activated in $proc_\alpha(x)$, e_{i_2} is activated in $proc_\alpha(x) \circ e_{i_1}$, ... , e_{i_k} is activated in $proc_\alpha(x) \circ e_{i_1} \circ e_{i_2} \circ \ldots \circ e_{i_{k-1}}$. The last process in this sequence is $proc_\omega(x)$. $\eta(x)$ should contain at least $Min(proc_{ON})$.

The definition of $\eta(x)$ is now obtained by

$$\eta(x) := Min(proc_{ON}) \cup \{proc_\alpha(x)\} \cup \{proc_\alpha(x) \circ e_{i_1} \circ e_{i_2} \circ \ldots \circ e_{i_j} \mid 1 \leq j \leq k\} \quad (5)$$

where $\{e_1.n_1, e_2.n_2, \ldots, e_k.n_k\} = \psi^{-1}(x)$ and $e_{i_1} < e_{i_2} < \ldots < e_{i_j}$ is ordered as $\phi_{ON}(e_1), \phi_{ON}(e_2), \ldots, \phi_{ON}(e_j)$ in $\widetilde{w}$, i.e. there is a subsequence $[\lambda, \phi_{ON}(e_1)], [\lambda, \phi_{ON}(e_2)], \ldots, [\lambda, \phi_{ON}(e_j)]$ in $\widetilde{w}$, such that $\phi_{ON}(e_i)$ corresponds to the n_i-th occurrence of e_i in $pr_2(\widetilde{w})$. All processes in $\eta(x)$ are ordered by $\preceq$. As $\eta(x) \neq \emptyset$ the lub $\sqcup\{\eta(x)\}$ always exist. Hence in expressions like $\sqcup\{\eta(x) \mid x \in {}^\bullet t_2\}$ the set $\eta(x)$ is to be replaced by its lub.

It remains to prove:

1. Each element of $\eta(x)$ is a subprocess of $proc_{ON}$.
2. $proc_{val}(\widetilde{w}) = (proc_{SN}, \eta)$.

ad 1.: By definition $proc_\alpha(x)$ is a causal net (as a particular subnet of a causal net), but not necessarily a subprocess of $proc_{ON}$. It remains to show that there are no "gaps", i.e. for each transition e_2 of $proc_\alpha(x)$ each transition e_1 satisfying $e_1 \leq_{ON} e_2$ should also be a transition of $proc_\alpha(x)$. This easily follows from the property FMP of ψ (Def. 5.1) which gives : $e_1 \leq_{ON} e_2 \Rightarrow \psi(e_1) \leq_{SN} \psi(e_2)$. We have assumed that e_2 is in $proc_\alpha(x)$, hence $\psi(e_2) \leq_{SN} x$ and $\psi(e_1) \leq_{SN} \psi(e_2) \leq_{SN} x$. From the definition of $A_\alpha(x)$ (equation (2)) we conclude that e_1 also belongs to $proc_\alpha(x)$. All processes of $\eta(x)$ are of the form $proc_{\alpha(x)} \circ e_1 \circ e_2 \circ \ldots \circ e_j$. As we will prove later e_i is activated in $proc_{\alpha(x)} \circ e_1 \circ e_2 \circ \ldots \circ e_{i-1}$ $(1 \leq i \leq j)$. Therefore all elements of $\eta(x)$ are subprocesses of $proc_{ON}$.

ad 2.: Assume $\widetilde{w}_{n-1} = [\alpha_1, \beta_1][\alpha_2, \beta_2] \ldots [\alpha_{n-1}, \beta_{n-1}] \in FS_{ref}(EOS)$ and $proc_{ref}(\widetilde{w}_{n-1}) = (proc_{SN}(\widetilde{w}_{n-1}), proc_{ON}(\widetilde{w}_{n-1}), \varphi(\widetilde{w}_{n-1}))$. We prove by induction on n that

$$proc_{val}(\widetilde{w}_{n-1}) = (proc_{SN}(\widetilde{w}_{n-1}), \eta). \quad (6)$$

where η is defined as before (equation (5)).

If $n = 1$ then $\widetilde{w}_{n-1} = [\lambda, \lambda]$, hence by Definition 3.3: $proc_{val}([\lambda, \lambda]) = (proc_{\mathbf{M_0}}, \eta_0)$ where $\eta_0(x) := proc_{\mathbf{m_0}}$ for all $x \in Min(proc_{\mathbf{M_0}})$. On the other hand we have $proc_{SN}(\widetilde{w}_{n-1}) = Min(proc_{SN}) = proc_{\mathbf{M_0}}$. For $x \in Min(proc_{SN})$ we obtain by equation (5): $\eta(x) = Min(proc_{ON}) \cup \{proc_\alpha(x)\} = Min(proc_{ON}) \cup cl(\emptyset) = Min(proc_{ON}) = proc_{\mathbf{m_0}}$, which implies the induction hypothesis (equation (6)) for $n = 1$.

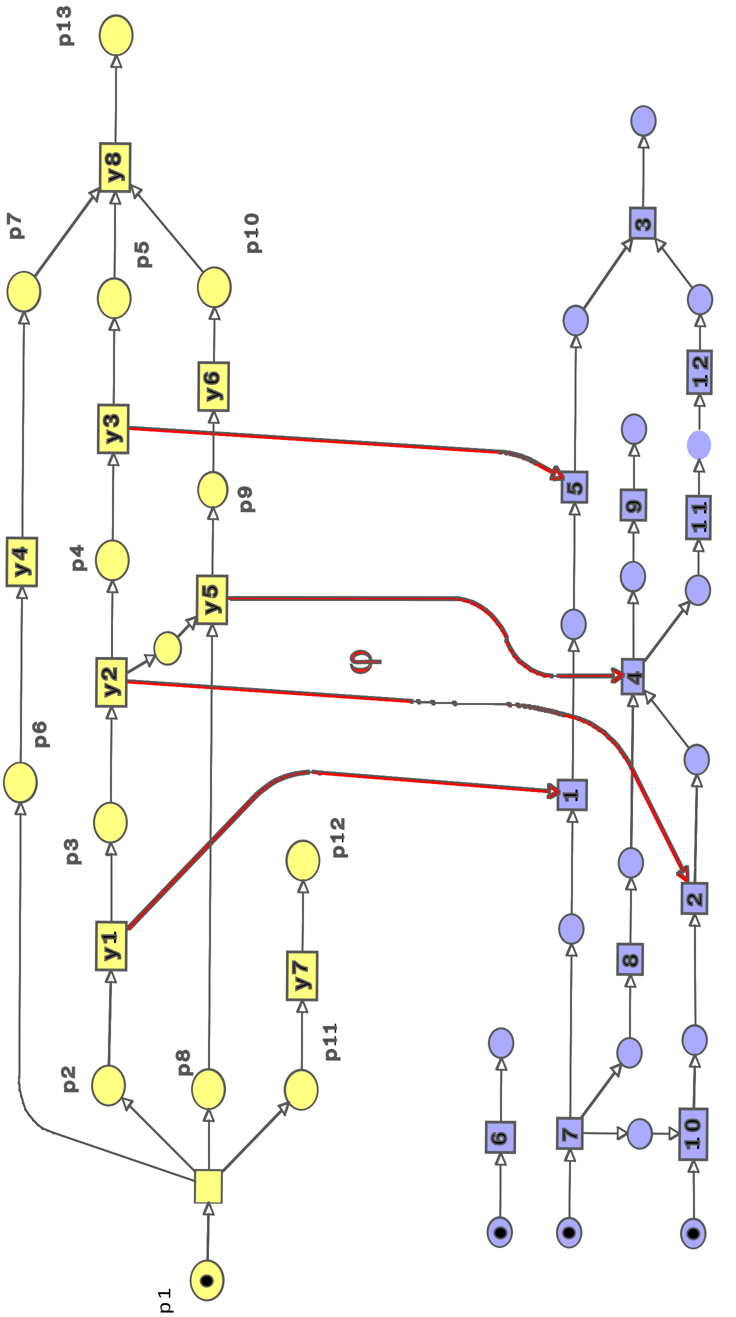

Figure 14: Elementary object system *WILD* with interaction mapping φ.

For the induction step assume $n \geq 1$ and $\widetilde{w}_n = \widetilde{w}_{n-1}[\alpha_n, \beta_n]$. We distinguish the following three cases:

a) $[\alpha_n, \beta_n] = [t, e], \;\; t \in P, e \in E$

b) $[\alpha_n, \beta_n] = [t, \lambda], \;\; t \in T$

c) $[\alpha_n, \beta_n] = [\lambda, e], \;\; e \in E$

case a): $[\alpha_n, \beta_n] = [t, e], \;\; t \in P, e \in E$

In this case, by $\widetilde{w}_n = \widetilde{w}_{n-1}[\alpha_n, \beta_n] \in FS_{ref}(EOS)$ transition t is activated in $proc_{SN}(\widetilde{w}_{n-1})$ and e is activated in $proc_{ON}(\widetilde{w}_{n-1})$, $\varphi(t) = e$, i.e. $proc_{SN}(\widetilde{w}_n) = proc_{SN}(\widetilde{w}_{n-1}) \circ t$ and $proc_{ON}(\widetilde{w}_n) = proc_{ON}(\widetilde{w}_{n-1}) \circ e$ exist. For the corresponding elements of these process enlargements $t_2 := \chi(proc_{SN}(\widetilde{w}_{n-1}), t)$ and $e_2 := \chi(proc_{ON}(\widetilde{w}_{n-1}), e)$ we have $\psi(e_2) = t_2$.

We have to show that the lub of all processes in the input places of t_2 activates e_2 (i.e. $\sqcup\{\eta(x) \mid x \in {}^\bullet t_2\}$ activates e_2) and $\sqcup\{\eta(x) \mid x \in {}^\bullet t_2\} \circ e_2 \in \eta(y)$ for the output places $y \in t_2{}^\bullet$.

The first part of this statement holds if each input place $b_1 \in {}^\bullet e_2$ is contained in a process of $\eta(x)$ for at least one input place $p_1 \in {}^\bullet t_2$. If ${}^\bullet b_1 = \emptyset$ then $b_1 \in Min(proc_{ON})$ which is contained in each process of any place of $proc_{val}(\widetilde{w}_{n-1})$ by construction. If ${}^\bullet b_1 \neq \emptyset$ then there is a transition $e_1 \in {}^\bullet b_1$ and we have $e_1 \leq_{ON} e_2$ (see the symbolic representation in Figure 12). By property FMP (Def. 5.1) of ψ it follows $\psi(e_1) \leq_{SN} \psi(e_2) = t_2$. There is an input place p_1 such that $\psi(e_1) \leq_{SN} p_1 \leq_{SN} t_2$. By the definition of η (equation (5)) $\eta(p_1)$ contains e_1 as desired.

We finish the discussion of case a) by showing that the definition of $\eta(p_2)$ (equation (5)) for the output places $p_2 \in t_2{}^\bullet$ coincides with the p-marking occurrence rule (Def. 2.5) and the related definition of a val-process (Def. 3.3). According to these definitions p_2 should contain as the input process the process that is generated (by the operation cl as in equation (1)) from the union of all transitions in processes in input places together with $\psi^{-1}(t_2) = e_2$ itself. As all lines passing through p_2 also meet the input places, this is in accordance with the definition of $\eta(x)$. More formally we compute: $proc_\alpha(p_2) \in \eta(p_2)$ with

$$proc_\alpha(p_2) = cl(A_\alpha(p_2))) \tag{7}$$

and

$$A_\alpha(p_2) = \{\psi^{-1}(y) \mid y \in Y_T \cup X_P \wedge y <_{SN} p_2\} = \{\psi^{-1}(t_2)\} \cup \bigcup_{p_1 \in {}^\bullet t_2} A_\alpha(p_1) \tag{8}$$

case b): $[\alpha_n, \beta_n] = [t, \lambda], \;\; t \in T$

This case is similar to case a). The only difference that $\psi^{-1}(t_2) = e_2$ is omitted in equation (8). In this case, by $\widetilde{w}_n = \widetilde{w}_{n-1}[\alpha_n, \beta_n] \in FS_{ref}(EOS)$ transition

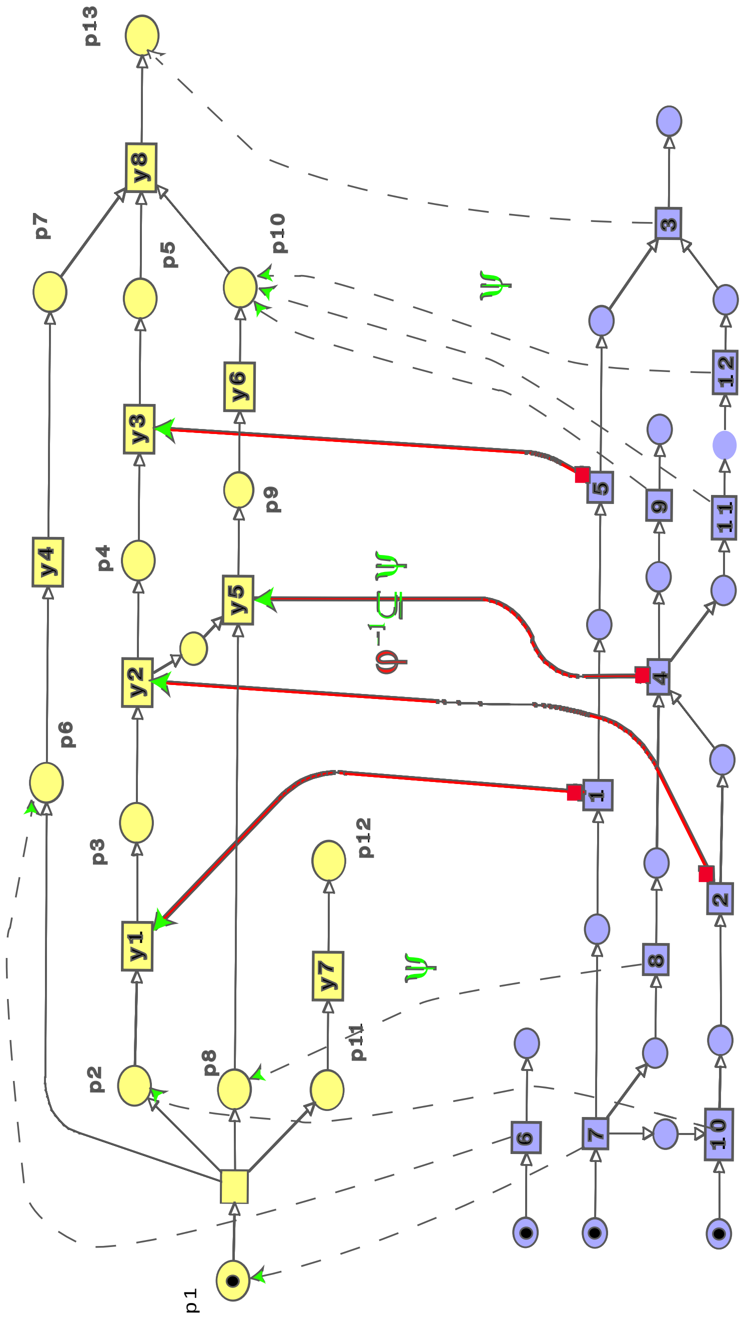

Figure 15: Elementary object system *WILD* with interaction mapping φ^{-1} and full extension ψ.

e is activated in $proc_{ON}(\widetilde{w}_{n-1})$, i.e. $proc_{ON}(\widetilde{w}_n) = proc_{ON}(\widetilde{w}_{n-1}) \circ e$ exist. Let be $\chi(proc_{ON}(\widetilde{w}_{n-1}), e) := e_2$ the corresponding element of the process enlargement.

case c): $[\alpha_n, \beta_n] = [\lambda, e], \ \ e \in E$

We will show that e_2 is activated in some process $proc \in \eta(p_1)$ where $p_1 = \psi(e_2)$. This is true if proc contains all input places $b_1 \in {}^\bullet e_2$ (see the symbolic representation in Figure 13). If ${}^\bullet b_1 = \emptyset$ then $b_1 \in Min(proc_{ON}(\widetilde{w}_{n-1})$ which is contained in each process of any place of $proc_{val}(\widetilde{w}_{n-1})$ by construction. If ${}^\bullet b_1 \neq \emptyset$ then there is a place $e_1 \in {}^\bullet b_1$ and we have $e_1 \leq_{ON} e_2$. By property FMP (Def. 5.1) of ψ it follows $\psi(e_1) \leq_{SN} \psi(e_2)$. By the definition of η (equation (5)) $\eta(p_1)$ and the discussion between equation (4) and equation (5) e_1 is contained in $proc_\omega(p_1)$. Hence there is some $1 \leq j \leq k$ such that $proc = proc_\alpha(p_1) \circ e_1 \circ e_2 \circ \ldots \circ e_j$ and $e_j = e_2$.

$\diamondsuit$

We illustrate the theorem for the occurrence sequence

$$\widetilde{w} = [\lambda, e_1], [t_1, \lambda], [t_3, e_3], [t_2, e_2], [t_4, e_4], [t_5, \lambda], [\lambda, e_5] \in FS_{ref}(con - task)$$

of the *EOS con-task* in Figure 2. The corresponding ref-process is shown in Figure 8. The mapping $\varphi^{-1} = \{(e_2.1, t_2.1), (e_3.1, t_3.1), (e_4.1, t_4.1)\}$ is fully extensible by $\{(e_1.1, p_1.1), (e_5.1, p_1.2)\}$, hence $\widetilde{w} \in FS_{val}(EOS)$. The val-process constructed in the proof of the theorem is given in Figure 9.

For better understanding the proof of Theorem 5.2, the EOS *WILD* in Figure 14 may be helpful. Both nets, the system net SN as well as the object net ON, are finite causal nets and thereby coincide with their maximal processes. Together with the given mapping φ they form a triple in rv-process-representation (Def. 3.4).

The inverse φ^{-1}, which is fully extensible. A possible extension is given in Figure 15 by dashed arrows. It may be checked that ψ satisfies the morphism property from part c) of Definition 5.1. To illustrate the construction of the map $\eta : X_P \to 2^{PROC(ON)}$ we give two elements of its value set: by the construction $\eta(p_{10})$ contains the process with the transition set $A_\alpha(p_{10}) := \{\psi^{-1}(y) \mid y \in Y_T \cup X_P \wedge y <_{SN} p_{10}\} = \{1, 2, 4, 7, 8, 10\}$. It also contains the process generated by the transitions of $A_\omega(p_{10}) := \{\psi^{-1}(y) \mid y \in Y_T \cup X_P \wedge y \leq_{SN} p_{10}\} = \{1, 2, 4, 7, 8, 10, 9, 11, 12\}$. The other elements of $\eta(p_{10})$ are those that lie between these two processes (w.r.t. the partial ordering $\prec$). To give another value of $\eta(x)$, the set $\eta(p_4)$ contains a single element, namely the process $cl(\{1, 2, 7, 10\})$. The full construction is given in Figure 16.

By the proof of Theorem 10 also a characterization of val-processes can be derived that is the analogon of Theorem 4.2. In a restricted form, namely when autonomous transitions are excluded, this result is published in [Val99b].

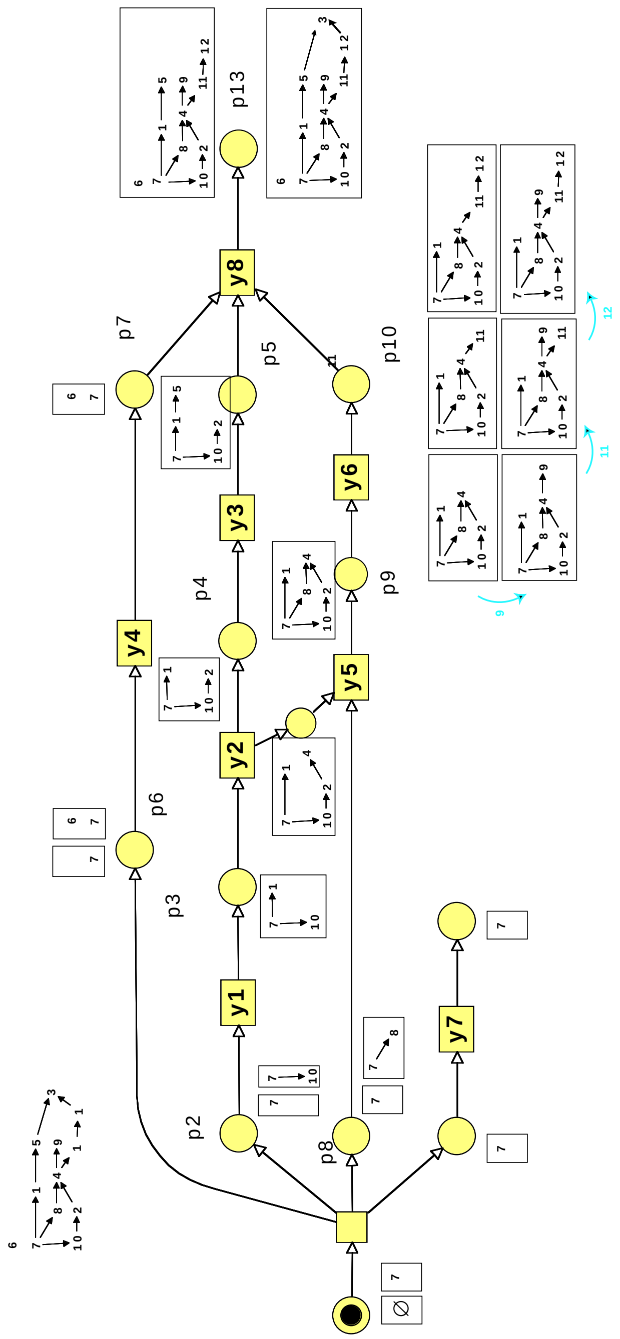

Figure 16: Full construction of the val-process for the EOS *WILD*.

6 From Value Semantics to Reference Semantics

In this section we discuss the reverse problem of the previous section, namely under which conditions an occurrence sequence with respect to the value semantics is also an occurrence sequence with respect to the reference semantics.

The occurrence sequence

$$\widetilde{w} = [\lambda, e_1], [t_1, \lambda], [t_2, e_2], [t_3, e_3], [t_4, e_3], [t_5, e_2], [t_6, e_4], [t_7, \lambda], [\lambda, e_5] \in FS_{val}(ser{-}task)$$

of the *EOS ser-task* in Figure 17 is *not* an occurrence sequence following the reference semantics (i.e. $\widetilde{w} \notin FS_{ref}(ser - task)$) as after the prefix $[\lambda, e_1], [t_1, \lambda], [t_2, e_2], [t_3, e_3]$ transition e_3 is not activated any more. The reason for this is that value semantics allows for "redundant execution" of object net transitions but reference semantics does not. Another source for a similar incompatibility is nondeterminism. To give an example consider the EOS *con-task-mod* from Fig. 7. Here the occurrence sequence $[\lambda, e_1], [t_1, \lambda], [t_2, e_2], [t_3, e_6] \in FS_{val}(EOS)$ is legal w.r.t. the value semantics but not for the reference semantics. The reason is that value semantics allows for different decisions of a conflict in different copies of the of object net, whereas w.r.t. reference semantics only one copy exists and the conflict cannot be resolved twice.

We conclude that a condition for an occurrence sequence with respect to the value semantics to be also an occurrence sequence with respect to the reference semantics is that the "global information" given in reference semantics has to be simulated. The notion of *least upper bound* gives some kind of consistency test on related copies of object nets. However, this test is restricted to the input places of a system net transition. In the given context it is therefore natural to extent the range of the lub-operation to all places in the system net.

Definition 6.1 *: Let $\mu : P \hookrightarrow PROC(ON)$ be a p-marking of a unary elementary object system $EOS = (SN, ON, \rho)$, where $SN = (P, T, W, \mathbf{M}_0)$ and $ON = (B, E, F, \mathbf{m}_0)$ and $e \in E$ is a transition of ON. Then μ is said to satisfy the* global consistency condition *(notation: $CONSIST(\mu)$) if $\bigsqcup \mu(P) := \bigsqcup\{\mu(p) \mid p \in P\}$ exists. The pair $(\mu, e.n)$ with $e \in E, n \in \mathbb{N}$ is said to satisfy the* global consistency condition *(notation: $CONSIST((\mu, e.n))$) if $CONSIST(\mu)$ and $(\bigsqcup \mu(P)) \circ e$ exists (and consequently $CONSIST((\bigsqcup \mu(P)) \circ e))$ and $\chi(\bigsqcup \mu(P), e) = e.n$.*

Recall, that in $\chi(proc, e) = e.n$ the expression $e.n$ denotes the name of the transition that extends the process $proc$ by the occurrence of the object net transition e. It is the n-th occurrence of e (see appendix).

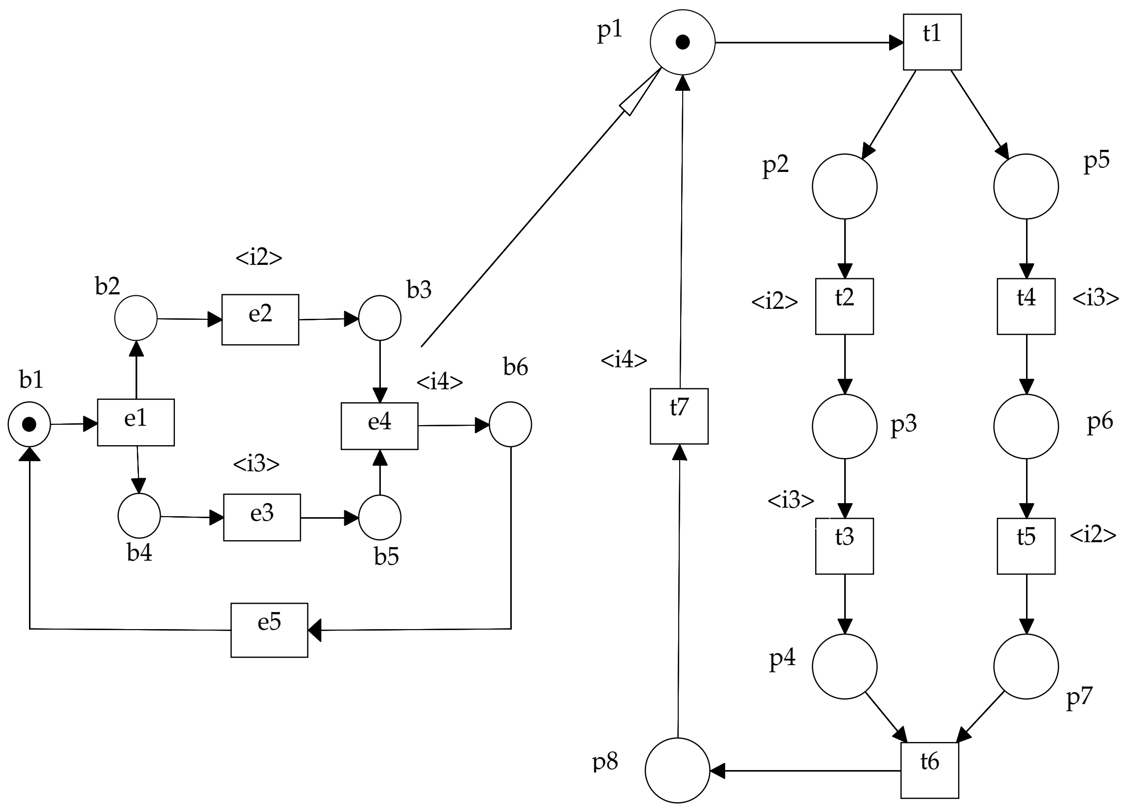

Figure 17: The Elementary object system ser-task.

Definition 6.2 *: Given an unary elementary object system EOS as in Definition 2.4, a system net transition t, an object net transition e and a p-marking μ. To define the strong successor marking relations $\mu \xrightarrow[\text{strg}]{[t,e]} \mu'$, $\mu \xrightarrow[\text{strg}]{[t,\lambda]} \mu'$ and $\mu \xrightarrow[\text{strg}]{[\lambda,e]} \mu'$ we proceed in three steps:*

a) Interaction*: $t \in T$, $e \in E$, $(t,e) \in \rho$*

$[t,e]$ strongly occurs *in μ and transforms μ into the successor p-marking μ' ($\mu \xrightarrow[\text{strg}]{[t,e]} \mu'$), if there is some $n \in \mathbb{N}$ with:*

1. $^\bullet t \subseteq dom\mu$ *and* $t^\bullet \cap dom\mu = \emptyset$,
2. $CONSIST(\mu, e.n)$ *holds,*
3. $\sqcup^\oplus t$ *and* $(\sqcup^\oplus t) \circ e$ *exist with* $\chi(\sqcup^\oplus t, e) = e.n$,
4. μ' *is defined by* $dom\, \mu' = (dom\, \mu \setminus {}^\bullet t) \cup t^\bullet$ *and for* $p \in dom\, \mu'$ *let be*

$$\mu'(p) := \begin{cases} (\sqcup^\oplus t) \circ e & \text{for } p \in t^\bullet \\ \mu(p) & \text{otherwise} \end{cases}$$

b) Transport*: $t \in T$, $t\rho = \emptyset$*

$[t,\lambda]$ strongly occurs *in μ and transforms μ into the successor p-marking μ' ($\mu \xrightarrow[\text{strg}]{[t,\lambda]} \mu'$), if*

1. $^\bullet t \subseteq dom\mu$ *and* $t^\bullet \cap dom\mu = \emptyset$,
2. $CONSIST(\mu)$ *holds,*
3. $\sqcup^\oplus t$ *exists,*
4. μ' *is defined by* $dom\, \mu' = (dom\, \mu \setminus {}^\bullet t) \cup t^\bullet$ *and for* $p \in dom\, \mu'$ *let be*

$$\mu'(p) := \begin{cases} \sqcup^\oplus t & \text{for } p \in t^\bullet \\ \mu(p) & \text{otherwise} \end{cases}$$

c) Object-autonomous event*: $e \in E$, $\rho e = \emptyset$*

$[\lambda,e]$ strongly occurs *in μ and transforms μ into the successor p-marking μ' ($\mu \xrightarrow[\text{strg}]{[\lambda,e]} \mu'$), if there are $p \in P$ and $n \in \mathbb{N}$ with:*

1. $CONSIST(\mu, e.n)$ *holds,*
2. $\mu(p) \circ e$ *exists with* $\chi(\mu(p), e) = e.n$,
3. μ' *is defined by* $dom\, \mu' := (dom\, \mu \setminus {}^\bullet t) \cup t^\bullet$ *and for* $p_1 \in dom\, \mu'$ *let be*

$$\mu'(p_1) := \begin{cases} \mu(p) \circ e & \text{for } p_1 = p \\ \mu(p_1) & \text{otherwise} \end{cases}$$

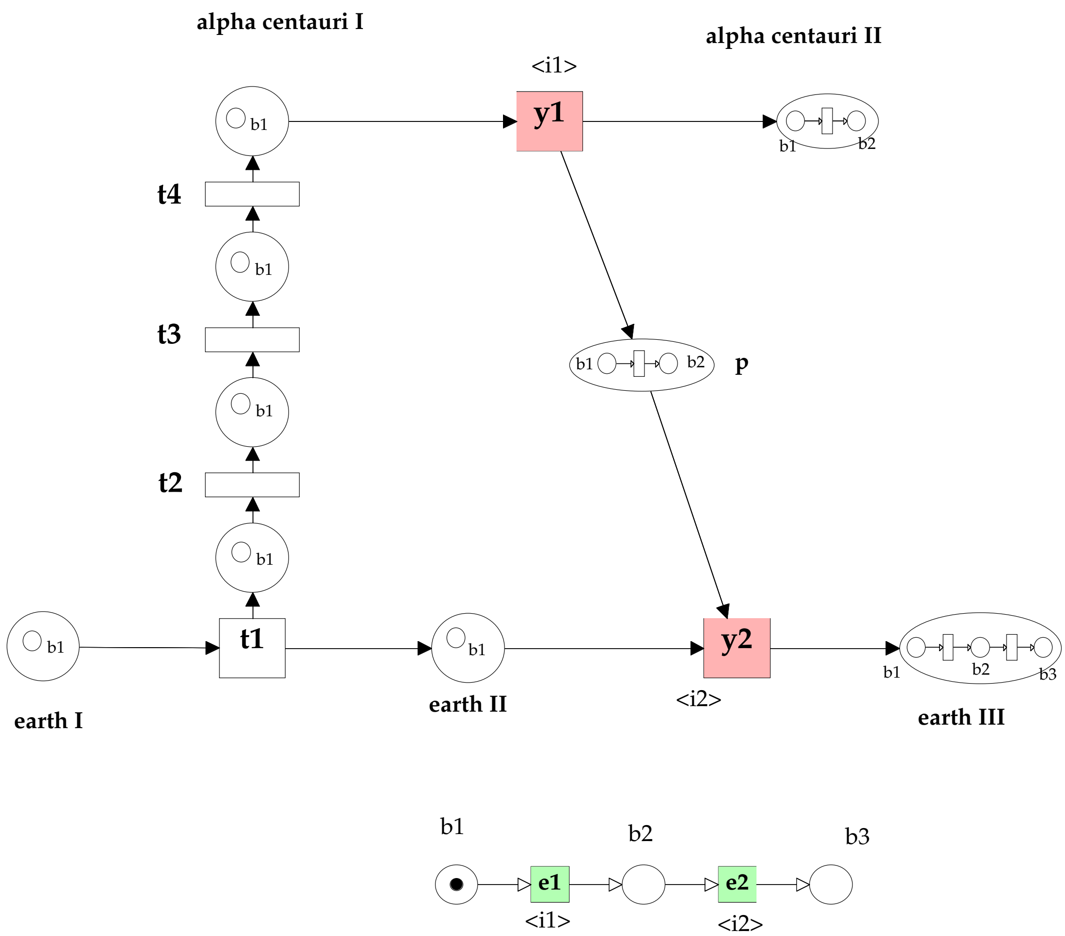

Figure 18: Elementary object system *alpha centauri extended.*

Definition 6.3 *The strong successor p-marking relation* $\mu \xrightarrow[\text{strg}]{[\alpha,\beta]} \mu'$ *is inductively extended to finite sequences* $\widetilde{w} \in \Gamma^*$ *(where* $\Gamma := (T \cup \{\lambda\}) \times (E \cup \{\lambda\}))\backslash[\lambda, \lambda]$ *and* $[\lambda, \lambda]$ *denotes the neutral element of the free monoid* Γ^**):*

- $\mu \xrightarrow{[\alpha,\beta]}_{strg} \mu$ *if* $[\alpha, \beta] = [\lambda, \lambda]$ *and*
- $\mu \xrightarrow{\widetilde{w}[\alpha,\beta]}_{strg} \mu'$ *if* $\exists \mu''.\ \mu \xrightarrow{\widetilde{w}}_{strg} \mu'' \wedge \mu'' \xrightarrow[\text{strg}]{[\alpha,\beta]} \mu'$ *for* $\widetilde{w} \in \Gamma^*$ *and* $[\alpha, \beta] \in \Gamma$

$FS_{strg} := FS_{strg}(EOS) := \{\widetilde{w} \in \Gamma^* \mid \exists \mu'.\ \mu_0 \xrightarrow{\widetilde{w}}_{strg} \mu'\}$ *denotes the set of* occurrence sequences (firing sequences) *of EOS with respect to the value semantics.*

Theorem 6.4 *Let EOS be a unary elementary object system and let* $\widetilde{w} \in FS_{val}(EOS)$ *be an occurrence sequence w.r.t. the value semantics. If* $\widetilde{w}$ *satisfies the strong value semantics occurrence rule i.e.* $\widetilde{w} \in FS_{strg}(EOS)$*, then* $\widetilde{w}$ *is also an occurrence sequence* $\widetilde{w} \in FS_{ref}(EOS)$ *w.r.t. the reference semantics.*

Proof

For the proof of this theorem the following mapping *strg* gives for any process-marking μ the corresponding bi-marking $(\mathbf{M}, \mathbf{m})$ by the maximal cut of the *lub* of all processes of μ:

$$strg : \mu \mapsto strg(\mu) := (\mathbf{M}, \mathbf{m}) := (dom\ \mu, Max(\sqcup\mu(P))) \tag{9}$$

Using this definition the theorem is proved by establishing the following implication, where $strg(\mu) \xrightarrow{[x,y]} strg(\mu')$ is the successor bi-marking relation of Definition 2.2:

$$\mu \xrightarrow[\text{strg}]{[x,y]} \mu' \qquad \Rightarrow \qquad strg(\mu) \xrightarrow{[x,y]} strg(\mu') \tag{10}$$

For the proof of this statement, the details of which are omitted, it is important to note that for each occurrence in the cases of a) (interaction) and c) (autonomous) the process $Max(\sqcup\mu(P)$ is effectively enlarged, which is fulfilled the definition of the strong occurrence rule (Definition 6.2), in particular by the conditions a) 3. and 4. as well as c) 2. and 3. By these conditions the enlargement of the process is performed with the same copy of the transition e that satisfies the global consistency condition i.e. $CONSIST(\mu, e.n) = true$. There are examples where confusion is possible. ♢

To give some examples consider the Alpha Centauri example again. The occurrence sequence $\widetilde{w}_1 := [t_1, \lambda], [t_2, \lambda], [t_3, \lambda], [t_4, \lambda], [y_1, e_1] \in FS_{val}$ (see Figure 3) is also strong, i.e. $\widetilde{w}_1 \in FS_{strg}$, and therefore legal w.r.t. the reference semantics: $\widetilde{w}_1 \in FS_{ref}$. This is, however, not the case for its prologation $\widetilde{w}_2 := \widetilde{w}_1[y_2, e_2] \notin FS_{strg}$ which is is not valid w.r.t. the strong occurrence rule. Now let us extend the EOS in such a way that there is "a message back to

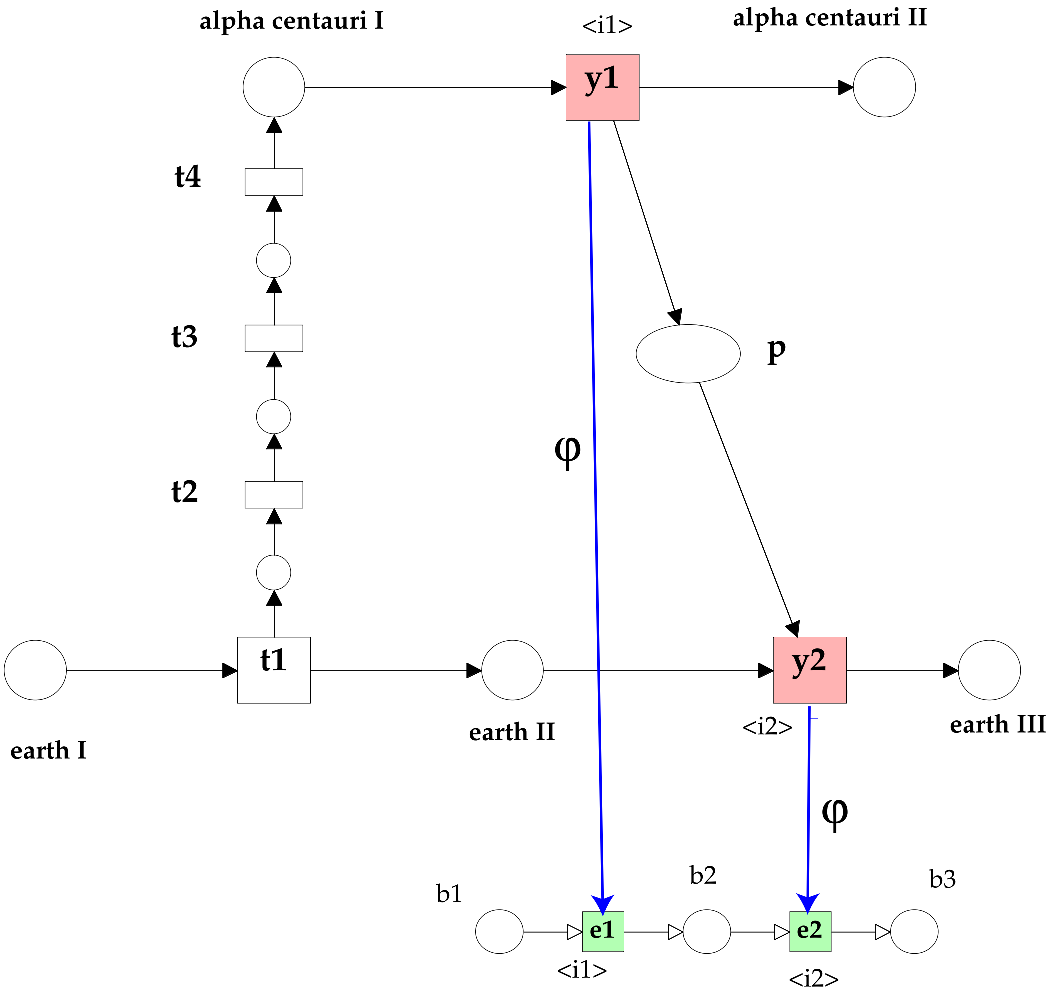

Figure 19: Processes of Figure 18 in rv-representation.

earth" as shown in Figure 18. This figure gives a process representation of the occurrence sequence $\widetilde{w}_3 := [t_1, \lambda], [t_2, \lambda], [t_3, \lambda], [t_4, \lambda], [y_1, e_1], [y_2, e_2] \in FS_{val}$. As it is valid w.r.t. the strong occurrence rule ($\widetilde{w}_3 \in FS_{strg}$) it is also valid w.r.t. the reference semantics ($\widetilde{w}_3 \in FS_{ref}$).

In Figure 19 the corresponding rv-representation is shown. As the map φ^{-1} is fully extensible also Theorem 5.2 can be applied. Hence this example provides an instance where both Theorems 5.2 and 6.4 hold, i.e. both transitions from value to reference semantics and vice versa are valid.

7 Conclusion

With this paper the semantics of object nets is further elaborated. Reference and value semantics are introduced, motivated and formally defined. Besides interleaving representations by occurrence sequences also a partial order representation by processes is given for both semantics. A characterizing condition for processes w.r.t the reference semantics is derived. A corresponding result for processes in value semantics has been established by the author in an earlier paper.

The relation of both semantics is studied by showing under which conditions one may pass from one to the other. While the proofs are based on partial orders the behavioral equivalence is stated by using occurrence sequences, which serve as common description language.

Given an object net behavior w.r.t. the reference semantics the transition to value semantics requires the distribution of state information to different places. This is done by using the notion of morphism that preserves the causal dependencies. In the literature such morphisms are used to formalize the simulation of one system by an other one and the given condition could be interpreted in a similar way. It says that the causal dependencies of the object net are to be preserved by the system net. In other words, the executing functional units cannot be more concurrent than the executed tasks.

Such a relation is not needed for reference semantics as there is a global representation of the current object net marking. Contrary to the previous case the passing from value semantics to reference semantics does not require the distribution of process information but a check on their consistency. This is formalized by the strong occurrence rule. An implementation of this semantics allows for a distributed storage of data but requires a global consistency check. This is similar to large scale distributed data bases where each modifying access is protected by consistency tests.

The results, obtained in this paper, will enable us to formulate a common model for object nets that includes both, reference and value semantics. In such a model a partition on the set of all places of the system net is assumed, reflecting the locality structure like in Figure 1, i.e. all places that can be associated to a node of a network are considered to be in the same class of the partition. Then for the transition rule we apply the strong occurrence rule with respect to all places in such a class and the value semantics occurrence rule for places in different classes. The case of reference semantics, as studied in this paper, becomes a special case, where the partition consists in only one class. In the same way value semantics is obtained if each place in its own class. Such a model would unify the theory and reflect the real world situation as discussed in connection with Figure 1.

8 Appendix: Processes and other formal notations

For a binary relation $R \subseteq A \times B$ the domain is defined by $dom R := \{a \in A \mid \exists b \in B.\ (a,b) \in R\}$ and the range by $range\ R := \{b \in B \mid \exists a \in A.\ (a,b) \in R\}$. This notion is also used with brackets: $dom(R), range(R)$. The definition is also used for relations which are a mapping $f : A \to B$. A partial mapping is denoted by $f : A \hookrightarrow B$, i.e. a mapping from $dom\ f$ to B. For a partial mapping $f : A \hookrightarrow B$ and an element $x \in A \backslash dom\ f$ the union $f \cup (x,y)$ is defined as an extension $g : A \hookrightarrow B$ by $dom\ g := dom(f) \cup \{x\}$ and $g(a) :=$ **if** $a = x$ **then** y **else** $f(a)$ **fi** for $a \in dom\ g$.

The non-sequential behavior of EN systems is given by causal nets (occurrence nets, cf [GR83], [BM85], [Roz87]). A process of an EN system $EN = (B, E, F, C)$ is defined by a node-labeled causal net $proc_{EN} = (X_B, Y_E, Z_F, \phi)$ such that $\phi : X_B \cup Y_E \to B \cup E$ satisfies

- $\phi(X_B) \subseteq B \wedge \phi(Y_E) \subseteq E$
- $\forall x_1, x_2 \in X_B : \phi(x_1) = \phi(x_2) \Rightarrow x_1 < x_2 \vee x_2 < x_1 \vee x_1 = x_2$
  (ϕ is injective on every B-cut of $proc_{EN}$)
- $\forall y \in Y_E : \phi({}^\bullet y) = {}^\bullet\phi(y) \wedge \phi(y^\bullet) = \phi(y)^\bullet$
- $\phi(Min(proc_{EN})) = C$

As usual, in this definition for an element $x \in B \cup E$ the set of *input elements* is the set ${}^\bullet x := \{y \in B \cup E \mid (y,x) \in F\}$ and $x^\bullet := \{y \in B \cup E \mid (x,y) \in F\}$ is the set of *output elements* of x. Furthermore in this definition $Min(proc_{EN}) :=$

$\{x \in X_B \mid {}^\bullet x = \emptyset\}$ is the set of *minimal elements* of the process. In the same way $Max(proc_{EN}) := \{x \in X_B \mid x^\bullet = \emptyset\}$ is the set of maximal elements. In this paper we always assume $E \subseteq range(F)$ (see [GR83]) and all processes to be finite. It follows that the set $Min(proc_{EN})$ is not empty. If $Min(proc_{EN}) = proc_{EN}$, then $proc_{EN}$ is called *initial process*. Since it consists of a set of places, in bijection with C, it is denoted by $proc_C$. The partial order of *causality* is denoted by $< := F^+$ and $\leq := F^*$ is the reflexive closure. If not excluded explicitly, a unique naming scheme is used for the places and transitions $X_B \cup Y_E$ of $proc_{EN}$ by defining $\phi^{-1}(x) := \{x.1, x.2, \ldots, x.k\}(x \in B \cup E, k = |\phi^{-1}(x)|)$ such that $x.1 < x.2 < \ldots < x.k$. Intuitively, $x.i$ denotes the i-th occurrence of x in the causal net $proc_{EN}$. This is well-defined, as each set $\phi^{-1}(x)$ is totally ordered by the causal relation $<$.

With li $:= \leq \cup \leq^{-1}$ and co $:= (X \times X)\backslash li \cup id_X$ (where $X := X_B \cup Y_E$). $c \subseteq X$ is a $co - set$ (anti-chain) iff $\forall x, y \in c : x \,\mathrm{co}\, y$. c is a *cut* if it is a maximal co-set (i.e. $\forall z \in X\backslash c\,.\exists y \in c.\ z \,\mathrm{li}\, y$); the set of cuts of $proc_{EN}$ will be denoted by $\mathcal{C} = \mathcal{C}(proc_{EN})$. A cut containing only elements of X_B (i.e. $c \subseteq X_B$) is a *B-cut* and $\mathcal{BC}(proc_{EN})$ denotes the set of all B-cuts of $proc_{EN}$.

For $A \subseteq X$ let $\downarrow A := \{y \in X \mid \exists x \in A.\ y \leq x\}$ and $\uparrow A := \{y \in X \mid \exists x \in A.\ \leq y\}$. A partial order is defined on the on the set $PROC(EN)$ of all processes of EN as follows. Given two processes $proc_1 = (X_1, Y_1, Z_1, \phi_1)$ and $proc_2 = (X_2, Y_2, Z_2, \phi_2)$ of EN, we define $proc_1 \preceq proc_2$ ($proc_1$ is an initial subprocess of $proc_2$) by $\exists A \in \mathcal{BC}(proc_2) : X_1 = X_2 \cap \downarrow A, Y_1 = Y_2 \cap \downarrow A, Z_1 = Z_2 \cap A^{\downarrow}$and $\phi_1 = \phi_{2|A^{\downarrow}}$, where $A^{\downarrow} := \downarrow A \times \downarrow A$. The preceding definitions are similar to an introduction in [BM85]. The notion of *lub* is used there to define infinite processes. Here, we are more interested in least upper bounds of finite sets of finite processes.

For any two processes $proc_1, proc_2 \in PROC(EN)$ the set $\mathcal{PR} := \{proc \in PROC(EN) \mid proc \preceq proc_1 \wedge proc \preceq proc_2\}$ is non-empty (since $proc_C \in \mathcal{PR}$) and contains an unique maximal element (since for B-cuts $Max(proc_i)$ and $Max(proc_j)$ also $\downarrow(Max(proc_i) \cap Max(proc_j))$ defines an initial subprocess of $proc_1$ and $proc_2$). This unique process is denoted by $proc_1 \sqcap proc_2$. In the "union" of two processes $proc_1 \cup proc_2 := (X_1 \cup X_2, Y_1 \cup Y_2, Z_1 \cup Z_2, \phi_1 \cup \phi_2)$ the elements of $proc_1 \sqcap proc_2$ are identified by the naming convention. In this construction the union $\phi_1 \cup \phi_2$ is understood as an union of the corresponding relations and is also a well-defined map again. $proc_1 \cup proc_2$ is not necessarily a process, however.

Let $\mathcal{PR} = \{proc_i \mid proc_i = (X_i, Y_i, Z_i, \phi_i), i \in I\}$ be a finite set of processes of EN. A process $proc_0$ such that $proc_i \preceq proc_0$ for all $i \in I$ is said to be an *upper bound* of $\mathcal{PR}$. Then, with respect to the order $\preceq$, there exists a *least upper bound* (lub) $\sqcup(\mathcal{PR})$ of $\mathcal{PR}$. Define the net union

$\bigcup(\mathcal{PR}) := (\cup_{i\in I}X_i, \cup_{i\in I}Y_i, \cup_{i\in I}Z_i, \cup_{i\in I}\phi_i)$ by extending the union of processes introduced above. Then $\mathcal{PR}$ has a lub iff $\bigcup(\mathcal{PR})$ is a process of EN. In this case $\sqcup(\mathcal{PR}) = \bigcup(\mathcal{PR})$.

Next we define the prolongation of a process by a transition occurrence. Let $proc_{EN} = (X_B, Y_E, Z_F, \phi)$ be a process of the (contact free) EN system $EN = (B, E, F, C)$ and let $e \in E$ be a transition, that has concession in the marking $\mathbf{m} = \phi(Max(proc_{EN}))$. Then in $\{proc \mid proc \in PROC(EN), proc_{EN} \preceq proc\}$ there is a unique process having $Y_E \cup \{y\}$ as a set of transitions, where $y \notin Y_E$ is a new element with $\phi(y) = e$. The new element y is denoted by $\chi(proc_{EN}, e)$. Its identifier is unique by the naming convention and has the form $e.n$ for some $n \in \mathbb{N}$. This process is the *prolongation* of $proc_{EN}$ by e and is denoted by $proc_{EN} \circ e$. For any occurrence sequence $w = e_1e_2 \ldots e_k \in F(EN)$ the process $proc(w) := proc_C \circ e_1 \circ e_2 \circ \ldots \circ e_k \in PROC(EN)$ is said to be the process corresponding to w. (Recall that $proc_C$ is the initial process of EN.) Vice versa, w is obtained from $proc(w)$ as $w = \phi(y_{i_1})\phi(y_{i_2})\ldots\phi(y_{i_k})$ by choosing on the set $\{y_1, y_2, \ldots, y_k\}$ of transitions of $proc(w)$ a total order $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$, such that $y_{i_p} < y_{i_q}$ implies $p < q$ for all $p, q \in \{1, \ldots, k\}$. The resulting total order is denoted by $y_{i_p} <_w y_{i_q}$.

References

[AMVW99] W.v.d. Aalst, D. Moldt, R. Valk, and F. Wienberg. Enacting Interorganizational Workflows Using Nets in Nets. In J. Becker, M. zur Mühlen, and M. Rosemann (Eds.), editors, *Proceedings of the 1999 Workflow Management Conference on Workflow-based Applications, Münster, Nov. 9th 1999*, Working Paper Series of the Department of Information Systems, pages 117–136, University of Münster, Department of Information Systems, Steinfurter Str. 109, 481149 Münster, 1999. Working Paper No. 70.

[BM85] E. Best and A. Merceron. Frozen Tokens and D-Continity: A Study in Relating Systems Properties to Process Properties. In G. Rozenberg, editor, *Advances in Petri Nets*, volume 188 of *LNCS*, pages 48–61. Springer-Verlag, Berlin, 1985.

[Far99] B. Farwer. A Linear Logic View of Object Petri Nets. *Fundamenta Informaticae*, 37:225–246, 1999.

[GR83] U. Golz and W. Reisig. The Non-sequential Behaviour of Petri Nets. *Information and Control*, 57:125–147, 1983.

[Kum98] O. Kummer. Simulating Synchronous Channels and Net Instances. In *5. Workshop Algorithmen und Werkzeuge für Petrinetze*, Forschungsbericht Nr. 694, Fachbereich Informatik, pages 73–78, Universität Dortmund, 1998.

[KW99] O. Kummer and F. Wienberg. Renew - the Reference Net Workshop. *Petri Net Newsletter*, 56:12–16, 1999. see also: `http://www.renew.de/`.

[MV00] D. Moldt and R. Valk. Object-oriented Petri Nets in Business Process Modelling. In W. v. d. Aalst, J. Desel, and A. Oberweis, editors, *Business Process Management: Models, Techniques, and Empirical Studies*, to be published in Lecture Notes in Computer Science, Berlin, Heidelberg, New York, 2000. Springer.

[Ra91] J. Rumbaugh and al. *Object-Oriented Modeling and Design.* Prentice-Hall, London, 1991.

[Roz87] G. Rozenberg. Behaviour of elementary net systems. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models* *and their Properties*, number 254 in LNCS, pages 60–94. Springer-Verlag, Berlin, 1987.

[Thi87] P.S. Thiagarajan. Elementary Net System. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models and their Properties*, number 254 in LNCS, pages 26–59. Springer-Verlag, Berlin, 1987.

[Val87a] R. Valk. Modelling of task flow in systems of functional units. Bericht 124, Fachbereich Informatik, Universität Hamburg, 1987.

[Val87b] R. Valk. Nets in computer organisation. In W. Brauer, W. Reisig, and G. Rozeberg, editors, *Petri Nets: Central Models and their Properties*, volume 255, pages 218–233. Springer-Verlag, Berlin, 1987.

[Val98] R. Valk. Petri Nets as Token Objects - An Introduction to Elementary Object Nets. In J. Desel and M. Silva, editors, *Application and Theory of Petri Nets*, number 1420 in LNCS, pages 1–25, Berlin, 1998. Springer-Verlag.

[Val99a] R. Valk. Reference and Value Semantics for Object Petri Nets. In H. Weber, H. Ehrig, and W. Reisig, editors, *Colloquium on Petri Net Technologies for Modelling Communication Based Systems*. Fraunhofer Institute for Software and Systems Engineering ISST, Berlin, 1999.

[Val99b] R. Valk. Concurrency in Communicating Object Petri Nets. In F.DeCindio, G.A.Agha, and G.Rozenberg, editors, *Concurrent Object-Oriented Programming and Petri Nets*, LNCS. Springer-Verlag, Berlin, to appear in 1999.