Algebraic High-Level Nets and Processes Applied to Communication Platforms

Karsten Gabriel

Bericht-Nr. 2010/14
ISSN 1436-9915

Algebraic High-Level Nets and Processes Applied to Communication Platforms*

Karsten Gabriel
Fraunhofer Institute for Open Communication Systems (FOKUS), Berlin, Germany

Abstract

Petri nets are well-known to model communication structures and algebraic specifications for modeling data types. Algebraic High-Level (AHL) nets are defined as integration of Petri nets with algebraic data types, which allows to model the communication structure and the data flow within one modelling framework. Transformations of AHL-nets – inspired by the theory of graph transformations – allow in addition to modify the communication structure. Moreover, high-level processes of AHL-nets capture the concurrent semantics of AHL-nets in an adequate way. Altogether we obtain a powerful integrated formal specification technique to model and analyse all kinds of communication based systems.

In this paper we give a comprehensive introduction of this framework. This includes main results concerning parallel independence of AHL-transformations and the transformation and amalgamation of AHL-occurrence nets and processes. Moreover, we show how this can be applied to model and analyse modern communication and collaboration platforms like Google Wave and Wikis. Especially we show how the Local Church-Rosser theorem for AHL-net tranformations can be applied to ensure the consistent integration of different platform evolutions. Moreover, the amalgamation theorem for AHL-processes shows under which conditions we can amalgamate waves of different Google Wave platforms in a compositional way.

1 Introduction

Algebraic specifications for data types have been proposed about 35 years ago in the US [Zil74, GTWW75, Gut75] and studied in more detail in Europe since 1980 [EKT+80, HKR80, CIP81]. The theory of concurrency, on the other hand, has its roots in Petri nets advocated in the PhD thesis of Petri [Pet62] already in 1962, but a break through from an algebraic point of view was the development of CCS by Milner [Mil80] and the concept of Petri nets as monoids by Meseguer and Montanari [MM90]. The stream processing functions developed by Broy [Bro85] can be seen as one of the first specification techniques for concurrent and distrubuted systems combining algebraic data type and process techniques. High-level nets based on low-level Petri nets [Pet62, Roz87, Rei85] and data types in ML have been studied as coloured Petri nets by Jensen [Jen91] and – using algebraic data types – as algebraic high-level (AHL) nets in [Rei90, PER95, ER97].

Inspired by the theory of graph transformations [Ehr79, Roz97] transformations of AHL-nets were first studied in [PER95] which – in addition to the token game – also allow to modify the net structure by rule based transformations. In this paper we consider parallel independence of AHL transformations leading to a Local Church-Rosser Theorem for Petri net transformations. The concept of processes in Petri nets is essential to model not only sequential, but especially concurrent firing behaviour. A process of a low-level Petri net N is given by an occurrence net K together with a net morphism $p : K \to N$. Processes of high-level nets AN are often defined as processes $p : K \to$

*This work has been supported by the Integrated Graduate Program on Human-Centric Communication at TU Berlin

$Flat(AN)$ of the corresponding low-level net $Flat(AN)$, called flattening of AN. However, this is not really adequate, because the flattening is in general an infinite net and the data type structure is lost. For this reason high-level processes for algebraic high-level nets have been introduced in [EHP+02, Ehr05], which are high-level net morphisms $p : K \to AN$ based on a suitable concept of high-level occurrence nets K. Inspired by the concept of amalgamation of processes for open low-level nets [BCEH01] we are able to define also amalgamation of high-level processes and a corresponding theorem in this paper, which shows under which conditions processes can be composed and decomposed leading to a compositional process semantics.

The main aim of this paper is to give a comprehensive introduction to the integrated framework of transformations of AHL-nets and amalgamation of high-level processes and to show how this can be applied to modern communication platforms. The modeling of Skype with transformations of high-level nets has been demonstranted in [MEE+10] already. In this paper we show how our integrated framework can be used to model basic aspects of Google Wave [Goo10b] and wiki-based systems. As detailed case studies we introduce in Section 2 Google Wave, which also used as running example for the following sections.

In Section 3 we introduce AHL-nets together with high-level processes in the sense of [Ehr05]. Rule based transformations in analogy to graph transformation systems [Roz97] are introduced in Section 4 for AHL-nets and in Section 5 for AHL-processes and applied to the evolution of Google Wave communication platforms and waves. The Local Church-Rosser Theorem for AHL-net transformations in Section 6 can be applied to show under which conditions different evolutions are independent such that they can be integrated consistently. Amalgamation – including composition and decomposition – of high-level processes is studied in Section 7 and applied to the amalgamation of waves, which are considered as processes of Google Wave platforms. The conclusion in Section 8 includes a summary and future work.

2 Case Studies

In this section we introduce our main case study Google Wave and discuss shortly wiki-based systems as additional example of communication platforms which can be modeled using our integrated framework. The interesting fact on these case studies is that the result of the communication in contrast to email, text chat or forums does not grow in a linear way but it is possible to make changes on previous contributions. This makes it important to model not only the systems and the communicaton but also the possibly parallel history of the communication.

Google Wave is a communication platform developed by the company Google [Goo10a]. Although Google itself has stopped the development of Google Wave, the communication platform is now an open source product under the name "Wave in a box" which allows everyone to run and evolve own Google Wave servers and clients. We have chosen Google Wave as running example for this paper because it allows demanding modern features that can be expected to be found in many other communication systems, such as near-real-time communication. This means that different users can simultaneously edit the same document, and changes of one user can be seen almost immediately by the other users.

Note that we do not focus on the communication between servers and clients in this contribution but on the communication between users. For details on the modeling of the more technical aspects of the server-to-server and client-to-server communication we refer to [Yon10].

2.1 Google Wave

In Google Wave users can communicate and collaborate via so-called waves. A wave is like a document which can contain diverse types of data that can be edited by different invited users. The changes that are made to a wave can be simultaneously recognized by the other participating users. In order to keep track of the changes that have been made, every wave contains also a history of all the actions in that wave.

Moreover, Google Wave supports different types of extensions which are divided into gagdets and robots. The extensions are programs that can be used inside of a wave. The difference between gadgets and robots is that gadgets are not able to interact with their environment while robots can be seen as automated users that can independently create, read or change waves, invite users or other robots, and so on. This allows robots for example to do real-time translation or highlighting of texts that are written by different users of a wave. Clearly, it is intended to use different robots for different tasks and it is desired that multiple robots interact without conflicts. This makes the modeling and analysis of Google Wave very important in order to predict possible conflicts or other undesired behavior of robots.

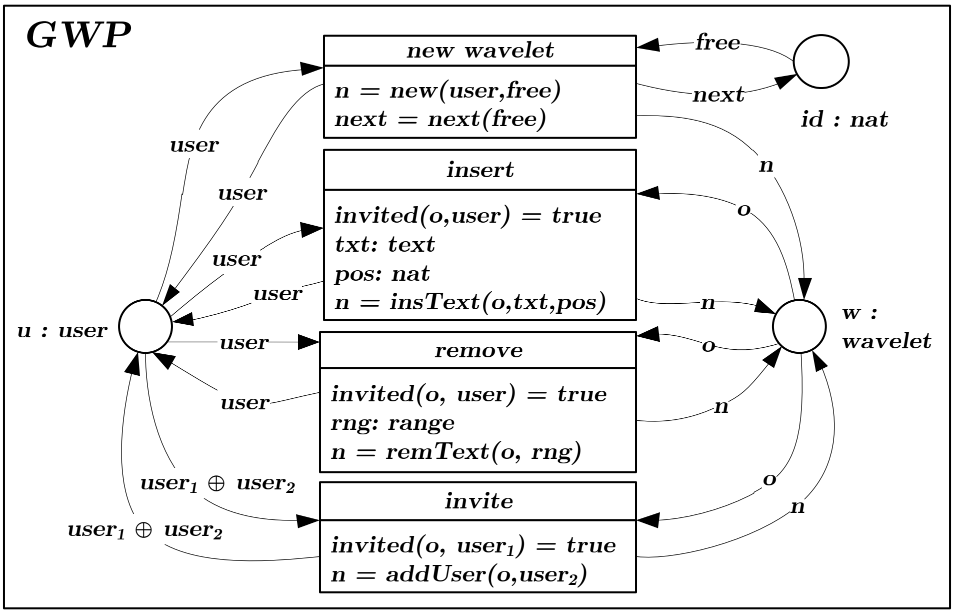


Figure 1: AHL-net *GWP* for a Google Wave platform

We claim that an adequate modeling technique for Google Wave are algebraic high-level (AHL) nets, which is an integration of the modeling technique of low-level Petri nets [Pet62, Roz87, Rei85] and algebraic data types. Figure 1 shows the structure of an AHL-net *GWP* which has 3 places and 4 transitions with firing conditions, where the pre and post arcs are labelled with variables of an algebraic signature. The AHL-net *GWP* models a Google Wave platform with some basic features like the creation of new waves, changes to existing waves, and the invitation of users to a wave which are modeled by the transitions *new wavelet*, *insert*, *remove* and *invite*.

A wavelet is a part of a wave that contains a user ID, a list of XML documents and a set of users which are invited to modify the wavelet. For simplicity we model in our example only the simple case that every wavelet contains only one single document and the documents contain only plain text. In order to obtain a more realistic model one has to extend the used algebraic data part of the model given by the signature Σ-$Wave$ shown in Table 2 and the Σ-$Wave$-algebra A in Table 3 of Section 3.

The transitions of the net contain firing conditions in the form of equations over the signature Σ-$Wave$. In order to fire a transition there has to be an assignment v of the variables in the environment of the transition such that the firing conditions are satisfied in the algebra A. The pair (t, v) is then called a consistent transition assignment. Moreover, there have to be suitable data elements in the pre domain of the transition. For example, in order to fire the transition *insert* we need a wavelet on the place w which can be assigned by the variable o and a user on the place u that can be assigned by the variable *user* such that the user is invited to the selected wavelet. Moreover, we need a text *txt*, a natural number *pos* and a new wavelet n such that n is the wavelet which is obtained by inserting the text *txt* on position *pos* into the original wavelet o.

The assignment v then determines a follower marking which is computed by removing the assigned data tokens in the pre domain of the transition and adding the assigned data tokens in the post domain. In the case of the transition *insert* this means that we remove the old wavelet from the place w and replace it by a new wavelet which contains the newly inserted text at the right position. For more details on the operational semantics of AHL-nets we refer to [Ehr05].

Due to the parallel semantics of the real-time communication in Google Wave a suitable modelling technique to capture the waves with their history, i. e. all states outgoing from their creation, are AHL-processes with instantiations which are introduced in [EHGP09]. Fig. 2 shows an example of an AHL-process *Wave* which abstractly models a wave that contains two wavelets created by possibly different users. A concrete instantiation *Inst* of the wave is shown in Fig. 3.

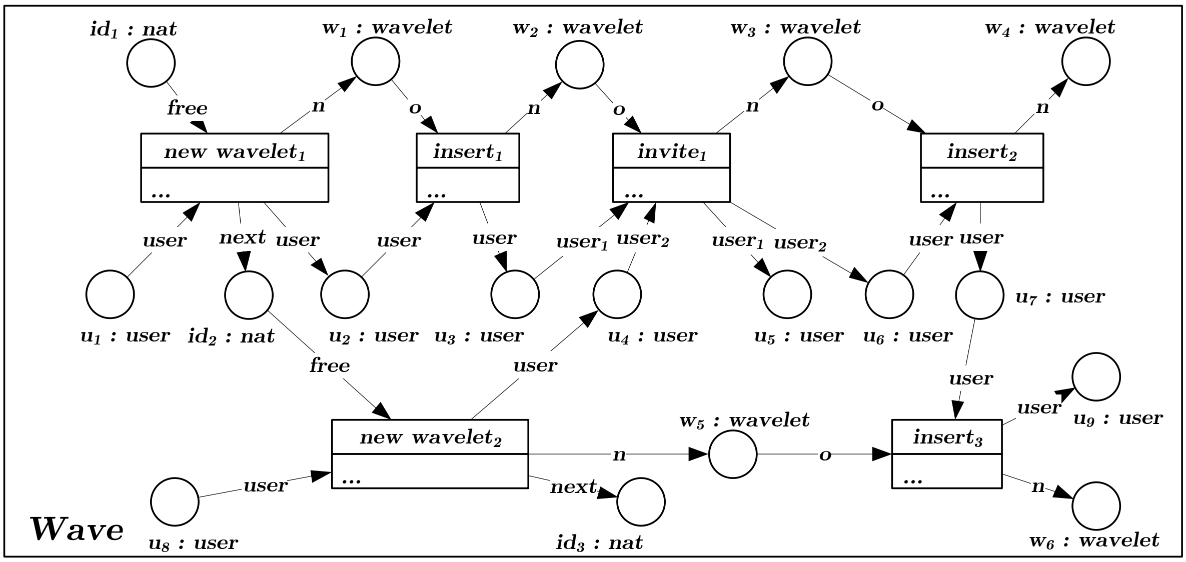


Figure 2: Abstract model *Wave* of a wave

The instantiation is a special low-level occurrence net with the same structure as the high-level occurrence net *Wave* that captures a part of the semantics of the high-level net, i. e. the places of the net *Inst* are markings of the places in the net *Wave* and the transitions of *Inst* correspond to firing steps of net *Wave* in the sense that they are pairs (t, v) of transitions t of the *Wave* together with an assignments v (see Table 1) such that (t, v) is a consistent transition assignment and the pre resp. post domain of (t, v) in *Inst* is the assigned pre resp. post domain of t under v in the net *Wave*. In fact, there can be different instantiations of one AHL-occurrence net, each one capturing one concurrent firing sequence in the platform GWP.

2.2 Wiki-Based Systems

A wiki is a website that manages information and data which can be easily created and edited by different users. Today wikis are used in many systems for communication, documentation, planning or other types of activities that involve the sharing of information. The most famous example of a wiki is the online encyclopedia Wikipedia. The collaboration via wikis is very similar to the one via Google Wave except for the near-real-time aspect. Due to the possibility to change contributions of other participants also the history of a wiki page is of importance and it is also usual to use automated scripts ("bots") for minor changes like the correction of common spelling mistakes. So also in the

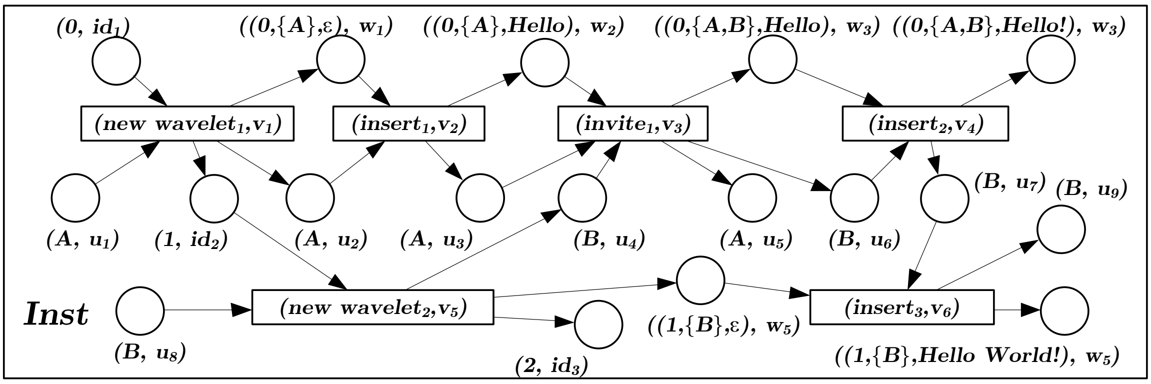


Figure 3: Concrete Instantiation $Inst$ of the $Wave$ Model

Table 1: Assignments of the Instantiation *Inst*

$v_1 : \{free, user, next, n\} \to A$
$free \mapsto 0, user \mapsto \texttt{A}, next \mapsto 1, n \mapsto (0, \{\texttt{A}\}, \epsilon)$
$v_2 : \{o, user, pos, txt, n\} \to A$
$o \mapsto (0, \{\texttt{A}\}, \epsilon), user \mapsto \texttt{A}, pos \mapsto 0, txt \mapsto \texttt{Hello}, n \mapsto (0, \{\texttt{A}\}, \texttt{Hello})$
$v_3 : \{user_1, user_2, o, n\} \to A$
$user_1 \mapsto \texttt{A}, user_2 \mapsto \texttt{B}, o \mapsto (0, \{\texttt{A}\}, \texttt{Hello}), n \mapsto (0, \{\texttt{A}, \texttt{B}\}, \texttt{Hello})$
$v_4 : \{o, user, pos, txt, n\} \to A$
$o \mapsto (0, \{\texttt{A}, \texttt{B}\}, \texttt{Hello}), user \mapsto \texttt{B}, pos \mapsto 5, txt \mapsto !, n \mapsto (0, \{\texttt{A}, \texttt{B}\}, \texttt{Hello!})$
$v_5 : \{free, user, next, n\} \to A$
$free \mapsto 1, user \mapsto \texttt{B}, next \mapsto 2, n \mapsto (1, \{\texttt{B}\}, \epsilon)$
$v_6 : \{o, user, pos, txt, n\} \to A$
$o \mapsto (1, \{\texttt{B}\}, \epsilon), user \mapsto \texttt{B}, pos \mapsto 0, txt \mapsto \texttt{Hello World!}, n \mapsto (1, \{\texttt{B}\}, \texttt{Hello World!})$

case of wiki-based systems it is relevant to have a process model of the system in order to analyse possible conflicts.

The example of a Google Wave platform given above can be modified to model a wiki-based system. This can be done by changing the transition *new wavelet* into a transition *new wiki page* with corresponding changes to the signature, algebra and the firing conditions of the transitions in order to model wiki pages (e. g. with an URL instead of an ID) instead of wavelets.

3 Algebraic High-Level Nets and their Processes

In the following we review the definition of AHL-nets and their processes from [Ehr05, EHP+02] based on low-level nets in the sense of [MM90], where $X^{\oplus}$ is the free commutative monoid over the set X. Note that $s \in X^{\oplus}$ is a formal sum $s = \sum_{i=1}^{n} \lambda_i x_i$ with $\lambda_i \in \mathbb{N}$ and $x_i \in X$ meaning that we have λ_i copies of x_i in s and for $s' = \sum_{i=1}^{n} \lambda'_i x_i$ we have $s \oplus s' = \sum_{i=1}^{n} (\lambda_i + \lambda'_i) x_i$.

Definition 3.1 (Algebraic High-Level Net)**.** An *algebraic high-level (AHL-) net*

$$AN = (\Sigma, P, T, pre, post, cond, type, A)$$

consists of

- a signature $\Sigma = (S, OP; X)$ with additional variables X;
- a set of places P and a set of transitions T;
- pre- and post domain functions $pre, post : T \to (T_\Sigma(X) \otimes P)^\oplus$;
- firing conditions $cond : T \to \mathcal{P}_{fin}(Eqns(\Sigma; X))$;
- a type of places $type : P \to S$ and
- a Σ-algebra A

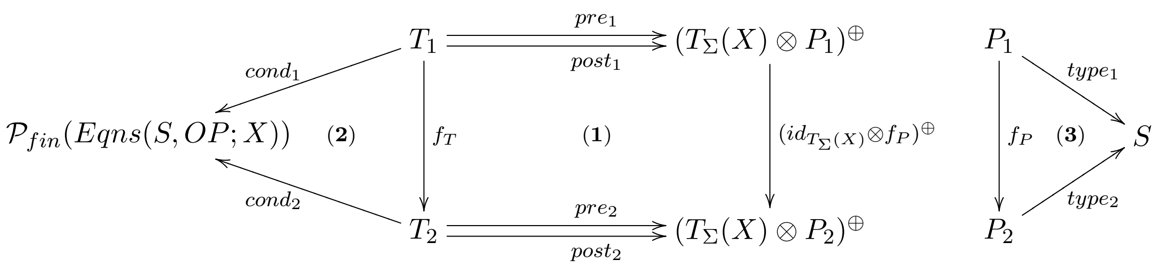
where the signature $\Sigma = (S, OP)$ consists of sorts S and operation symbols OP, $T_\Sigma(X)$ is the set of terms with variables over X,

$$(T_\Sigma(X) \otimes P) = \{(term, p) | term \in T_\Sigma(X)_{type(p)}, p \in P\}$$

and $Eqns(\Sigma; X)$ are all equations over the signature Σ with variables X.

An AHL-net morphism $f : AN_1 \to AN_2$ is given by $f = (f_P, f_T)$ with functions $f_P : P_1 \to P_2$ and $f_T : T_1 \to T_2$ satisfying

(1) $(id \otimes f_P)^\oplus \circ pre_1 = pre_2 \circ f_T$ and $(id \otimes f_P)^\oplus \circ post_1 = post_2 \circ f_T$,
(2) $cond_2 \circ f_T = cond_1$ and
(3) $type_2 \circ f_P = type_1$.



The category defined by AHL-nets (with signature Σ and algebra A) and AHL-net morphisms is denoted by **AHLNets** where the composition of AHL-net morphisms is defined componentwise for places and transitions.

Note that it is also possible to define a category of AHL-nets with different signatures and algebras which requires that the morphisms not only contain functions for places and transitions but also a signature morphism together with a general algebra morphism.

Definition 3.2 (Firing Behaviour of AHL-Nets)**.** A marking of an AHL-net AN is given by $M \in CP^\oplus$ where

$$CP = (A \otimes P) = \{(a, p) \mid a \in A_{type(p)}, p \in P\}$$

and $M = \sum_{i=1}^{n} \lambda_i(a_i, p_i)$ means that place $p_i \in P$ contains $\lambda_i \in \mathbb{N}$ data tokens $a_i \in A_{type(p_i)}$.

The set of variables $Var(t) \subseteq X$ of a transition $t \in T$ are the variables of the net inscriptions in $pre(t), post(t)$ and $cond(t)$. Let $v : Var(t) \to A$ be a variable assignment with term evaluation

$\overline{v} : T_\Sigma(Var(t)) \to A$, then (t, v) is a consistent transition assignment iff $cond_{AN}(t)$ is validated in A under v. The set CT of consistent transition assignments is defined by

$$CT = \{(t, v) | (t, v) \text{ consistent transition assignment}\}.$$

A transition $t \in T$ is enabled in M under v iff

$$(t, v) \in CT \text{ and } pre_A(t, v) \leq M$$

where $pre_A : CT \to CP^\oplus$ is defined by

$$pre_A(t, v) = \hat{v}(pre(t)) \in (A \otimes P)^\oplus$$

and

$$\hat{v} : (T_\Sigma(Var(t)) \otimes P)^\oplus \to (A \otimes P)^\oplus$$

is the obvious extension of $\overline{v}$ to sums of terms and places (similar $post_A : CT \to CP^\oplus$). Then the follower marking is computed by

$$M' = M \ominus pre_A(t, v) \oplus post_A(t, v).$$

Remark 3.1 (AHL-Morphisms Preserve Firing Behaviour). Given an AHL-net morphism $f : AN_1 \to AN_2$ the firing behaviour is preserved, i. e. for

$$M_1' = M_1 \ominus pre_{1,A}(t, asg) \oplus post_{1,A}(t, asg)$$

in AN_1 we have

$$M_2' = M_2 \ominus pre_{2,A}(f_T(t), asg) \oplus post_{2,A}(f_T(t), asg)$$

in AN_2 with $M_2 = \sum_{i=1}^{n} (a_i, f_P(p_i))$ for $M_1 = \sum_{i=1}^{n} (a_i, p_i)$ and similar M_2' constructed from M_1'.

Remark 3.2 (AHL-Nets with Individual Tokens). In contrast to the firing behaviour defined in Def. 3.2 it is also possible to define a marking over a set I of individuals and a marking function $m : I \to A \otimes P$ assigning each individual to a pair of a data element and a place. This makes it possible to distinguish the single tokens of a marking.

In order to fire a transition under a given marking it is then necessary to specify a token selection (M, m, N, n) where $M \subseteq I$ is the set of individuals which are consumed by the transition, N is a set of newly created individuals with $(I \setminus M) \cap N = \emptyset$ and $m : M \to A \otimes P$, $n : N \to A \otimes P$ are corresponding marking functions. If a selection together with a consistent transition assignment (t, asg) meets the *token selection condition*:

$$\sum_{i \in M} m(i) = pre_A(t, asg) \quad \text{and} \quad \sum_{i \in N} n(i) = post_A(t, asg)$$

then t is asg-enabled and the follower marking (I', m') can be computed by

$$I' = (I \setminus M) \cup N, \quad m' : I' \to A \otimes P \quad \text{with} \quad m'(x) = \begin{cases} m(x), \text{ if } x \in I \setminus M; \\ n(x), \text{ if } x \in N. \end{cases}$$

Although this *individual token approach* is more complicated than the *collective token approach* in Def. 3.2 it has some benefits like the possibity to formulate transformation rules which can not only change the net structure but also the marking of an AHL-net. For more details we refer to [MGE+10]. In this paper we still use the collective approach but we will also research processes of AHL-nets with individual tokens in the future.

Example 3.1 (Google Wave Platform)**.** The model of a Google Wave platform in Fig. 1 is an AHL-net

$$GWP = (\Sigma\text{-}Wave, P, T, pre, post, cond, type, A)$$

where the signature Σ-$Wave$ is shown in Table 2 and the Σ-$Wave$-algebra A is shown in Table 3. This signature and algebra is also used for all the following examples.

Let us consider the marking

$$M = (Alice, u) \oplus (Bob, u) \oplus (1, id) \oplus ((0, \{Alice, Bob\}, \epsilon), w)$$

which means that we have two users $Alice$ and Bob on the place u, a free ID 1 and an empty wavelet with ID 0 on place w where $Alice$ and Bob are invited. An assignment $asg : \{user, txt, pos, o, n\} \to A$ with $asg(user) = Alice$, $asg(txt) = Hello\ Bob$, $asg(pos) = 0$, $asg(o) = (0, \{Alice, Bob\}, \epsilon)$ and $asg(n) = (0, \{Alice, Bob\}, Hello\ Bob)$ satisfies the firing conditions of the transition *insert* and by firing the transition *insert* with assignment asg we obtain the follower marking

$$M' = (Alice, u) \oplus (Bob, u)\oplus, (1, id) \oplus ((0, \{Alice, Bob\}, Hello\ Bob), w)$$

where the assigned text $Hello\ Bob$ has been inserted at position 0 into the assigned wavelet.

Table 2: Signature Σ-$Wave$

sorts:	bool, nat , range, user, text, wavelet	
opns:	true, false : $\to$ bool	next : nat $\to$ nat
	start, end : range $\to$ nat	new : user nat $\to$ wavelet
	addUser : user wavelet $\to$ wavelet	invited : wavelet user $\to$ bool
	len : text $\to$ nat	sub : text range $\to$ text
	insText : wavelet text nat $\to$ wavelet	remText : wavelet range $\to$ wavelet

Now, we introduce AHL-occurrence nets based on low-level occurrence nets (see [GR83]) and AHL-processes according to [Ehr05, EHP[+]02]. The net structure of a high-level occurrence net has similar properties like a low-level occurrence net, but it captures a set of different concurrent computations due to different initial markings. In fact, high-level occurrence nets can be considered to have a set of initial markings for the input places, whereas there is only one implicit initial marking of the input places for low-level occurrence nets.

Definition 3.3 (AHL-Occurrence Net)**.** An AHL-occurrence net K is an AHL-net

$$K = (\Sigma, P, T, pre, post, cond, type, A)$$

such that for all $t \in T$ with $pre(t) = \sum_{i=1}^{n}(term_i, p_i)$ and notation $\bullet t = \{p_1, \dots, p_n\}$ and similarly $t\bullet$ we have

1. (*Unarity*): $\bullet t, t\bullet$ are sets rather than multisets for all $t \in T$, i. e. for $\bullet t$ the places $p_1 \dots p_n$ are pairwise distinct. Hence $|\bullet t| = n$ and the arc from p_i to t has a unary arc-inscription $term_i$.
2. (*No Forward Conflicts*): $\bullet t \cap \bullet t' = \emptyset$ for all $t, t' \in T, t \neq t'$
3. (*No Backward Conflicts*): $t \bullet \cap t' \bullet = \emptyset$ for all $t, t' \in T, t \neq t'$
4. (*Partial Order*): the causal relation $<_K \subseteq (P \times T) \cup (T \times P)$ defined by the transitive closure of $\{(p,t) \in P \times T \mid p \in \bullet t\} \cup \{(t,p) \in T \times P \mid p \in t\bullet\}$ is a finitary strict partial order, i. e. the partial order is irreflexive and for each element in the partial order the set of its predecessors is finite.

Table 3: Σ-$Wave$-algebra A

$A_{bool} = \{\mathtt{T}, \mathtt{F}\}$	$A_{nat} = \mathbb{N}$
$A_{user} = \{\mathtt{a}, \dots, \mathtt{z}, \mathtt{A}, \dots, \mathtt{Z}\}^*$	$A_{text} = \{\mathtt{a}, \dots, \mathtt{z}, \mathtt{A}, \dots, \mathtt{Z}, \dots\}^*$
$A_{wavelet} = A_{nat} \times \mathcal{P}(A_{user}) \times A_{text}$	$A_{range} = A_{nat} \times A_{nat}$

$true_A = \mathtt{T} \in A_{bool}$

$\mathrm{false}_A = \mathtt{F} \in A_{bool}$

$\mathrm{start}_A : A_{range} \to A_{nat}$ with $(s, e) \mapsto s$

$\mathrm{end}_A : A_{range} \to A_{nat}$ with $(s, e) \mapsto e$

$next_A : A_{nat} \to A_{nat}$ with $n \mapsto n + 1$

$new_A : A_{user} \times A_{nat} \to A_{wavelet}$ with $(u, id) \mapsto (id, \{u\}, \epsilon)$

$addUser_A : A_{user} \times A_{wavelet} \to A_{wavelet}$ with $(u, (id, uset, t) \mapsto (id, uset \cup \{u\}, t)$

$invited_A : A_{wavelet} \times A_{user} \to A_{bool}$ with $(u, (id, uset, t)) \mapsto \begin{cases} \mathtt{T} & \text{, if } u \in uset; \\ \mathtt{F} & \text{, else.} \end{cases}$

$len_A : A_{text} \to A_{nat}$ with $t \mapsto \begin{cases} 0 & \text{, if } t = \epsilon; \\ 1 + len_A(t_1 \dots t_n) & \text{, if } t = t_0 \dots t_n. \end{cases}$

$sub_A : A_{text} \times A_{range} \to A_{text}$ with

$(t, (s, e)) \mapsto \begin{cases} \epsilon & \text{, if } e < s \text{ or } len_A(t) \leq s; \\ t_s \dots t_n & \text{, if } t = t_0 \dots t_m, s \leq e, s < m \text{ and } n = min(m, e). \end{cases}$

$insText_A : A_{wavelet} \times A_{text} \times A_{nat} \to A_{wavelet}$ with

$((id, uset, t), nt, pos) \mapsto (id, uset, sub_A(t, (0, pos - 1)).nt.sub_A(t, (pos, len_A(t))))$

$remText_A : A_{wavelet} \times A_{range} \to A_{wavelet}$ with

$((id, uset, t), (s, e)) \mapsto (id, uset, sub_A(t, (0, s)).sub_A(t, (e, len_A(t))))$

AHL-occurrence nets (with signature Σ and algebra A) together with AHL-net morphisms between AHL-occurrence nets form the full subcategory **AHLONets** $\subseteq$ **AHLNets**.

Definition 3.4 (Input and Output Places)**.** Given an AHL-occurrence net K. We define the set $IN(K)$ of input places of K as

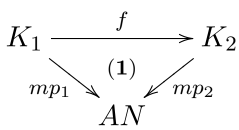
$$IN(K) = \{p \in P_K \mid \nexists\, t \in T_K : p \in t\bullet\}$$

and similar the set $OUT(K)$ of output places of K as

$$OUT(K) = \{p \in P_K \mid \nexists\, t \in T_K : p \in \bullet t\}$$

Note that due to the finitariness of AHL-occurrence nets there is always a nonempty set of input places $IN(K)$ whereas the set of output places $OUT(K)$, in general, may be empty.

Definition 3.5 (AHL-Process)**.** An AHL-process of an AHL-net AN is an AHL-net morphism $mp : K \to AN$ where K is an AHL-occurrence net.
The category **Proc(AN)** of AHL-processes of an AHL-net AN is defined as the full subcategory of the slice category **AHLNets** $\setminus AN$ such that the objects are AHL-processes. This means that the objects of **Proc(AN)** are AHL-process morphisms $mp : K \to AN$ and the morphisms of the category are AHL-net morphisms $f : K_1 \to K_2$ such that diagram (1) commutes.



Example 3.2 (Wave)**.** The abstract model *Wave* of a wave in Fig. 2 is an AHL-occurrence net. We can define a morphism $mp : Wave \to GWP$ which maps all places and transitions of the net *Wave* to the places and transitions, respectively, with the same name but without index. Then mp is an AHL-process w.r.t. the Google Wave platform GWP in Fig. 1.

Lemma 3.1 (AHL-Morphisms Reflect AHL-Occurrence Nets)**.** *Given an AHL-morphism* $f : K_1 \to K_2$. *If* K_2 *is an AHL-occurrence net then also* K_1.

Proof Idea. The unarity of K_2 together with the fact that AHL-morphisms preserve pre and post conditions imply that f maps only non-injectively on isolated places or transitions. Thus, K_1 basically has the same structural properties as K_1 which means that it does also satisfy the structural conditions to be an AHL-occurrence net. For a detailed proof see Appendix A.1. □

In order to define the restriction of AHL-processes via AHL-morphisms we need the following construction, which defines a pullback in the category **AHLNets**.

Definition 3.6 (Restriction of Algebraic High-Level Nets)**.** The *restriction* of an AHL-morphism $g_1 : AN_1 \to AN_3$ to AN_2 with an inclusion morphism $g_2 : AN_2 \to AN_3$ is given by $f_2 : AN_0 \to AN_2$ with inclusion $f_1 : AN_0 \to AN_1$, where AN_0 is constructed as subnet of AN_1 with $P_0 = g_{1,P}^{-1}(P_2) \subseteq P_1$ and $T_0 = g_{1,T}^{-1}(T_2) \subseteq T_1$ and f_2 is the restriction of g_1.

$$\begin{array}{ccc} AN_0 & \xrightarrow{f_1} & AN_1 \\ {\scriptstyle f_2}\downarrow & (\mathbf{PB}) & \downarrow{\scriptstyle g_1} \\ AN_2 & \xrightarrow[g_2]{} & AN_3 \end{array}$$

Remark 3.3 (Pullbacks). The diagram (PB) is a *pullback* diagram in the category **AHLNets**, i. e. (PB) is commutative and has the following universal property: For all AHL-nets AN_0' and AHL-morphisms $h_1 : AN_0' \to AN_1$, $h_2 : AN_0' \to AN_2$ with $g_1 \circ h_1 = g_2 \circ h_2$ there is a unique $h : AN_0' \to AN_0$ with $f_1 \circ h = h_1$ and $f_2 \circ h = h_2$. Moreover we can replace the inclusion $g_2 : AN_2 \to AN_3$ by an injection morphism $g_2 : AN_2 \to AN_3$ leading to an injective morphism $f_1 : AN_0 \to AN_1$.

Fact 3.2 (Extension and Restriction of AHL-Processes)**.**

1. ***Extension:*** *Given a process* $mp_1 : K_1 \to AN_1$ *and an AHL-morphism* $f : AN_1 \to AN_2$ *then* $mp_2 = f \circ mp_1 : K_1 \to AN_2$ *is a process of* AN_2 *called* extension of mp_1 along f.

2. ***Restriction:*** *Given a process* $mp_2 : K_2 \to AN_2$ *and an inclusion* $f : AN_1 \to AN_2$ *then the restriction* $mp_1 : K_1 \to AN_1$ *of* mp_2 *to* AN_1 *is a process of* AN_1 *with inclusion* $\phi : K_1 \to K_2$ *and pullback (PB).*

$$\begin{array}{ccc} K_1 & \xrightarrow{\phi} & K_2 \\ {\scriptstyle mp_1}\downarrow & (\mathbf{PB}) & \downarrow{\scriptstyle mp_2} \\ AN_1 & \xrightarrow[f]{} & AN_2 \end{array}$$

Proof.

1. Since K_1 is an AHL-occurrence net and $f \circ mp_1 : K_1 \to AN_2$ is an AHL-morphism it is also a process of AN_2.

2. Given process $mp_2 : K_2 \to AN_2$ and inclusion f the restriction $mp_1 : K_1 \to AN_1$ with inclusion $\phi : K_1 \to K_2$ is defined by Definition 3.6. Moreover, K_2 is an AHL-occurrence net which by Lemma 3.1 and AHL-morphism $\phi : K_1 \to K_2$ implies that K_1 is an AHL-occurrence net. Hence, $mp_1 : K_1 \to AN_1$ is a process of AN_1.

□

4 Transformation of Algebraic High-Level Nets

Due to the possibility to evolve the Google Wave platforms by adding, removing or changing features we need also techniques that make it possible to evolve the corresponding model of a platform. For this reason we introduce rule-based AHL-net transformations in the sense of graph transformations [EEPT06].

A production (or transformation rule) for AHL-nets specifies a local modification of an AHL-net. It consists of a left-hand side, an interface which is the part of the left-hand side which is not deleted and a right-hand side which additionally contains newly created net parts.

Definition 4.1 (Productions for AHL-Nets)**.** A *production for AHL-nets* is a span $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ of injective AHL-morphisms. We call L the left-hand side, I the interface, and R the right-hand side of the production p. In most examples l and r are inclusions.

$$L \xleftarrow{\;l\;} I \xrightarrow{\;r\;} R$$

Example 4.1 (Production for Platforms)**.** Fig. 4 shows an example of a production $p_1 : L_1 \overset{l_1}{\leftarrow} I_1 \overset{r_1}{\rightarrow} R_1$ where the morphisms l_1 and r_1 are inclusions. The production can be used to replace two transitions *insert* and *remove* by a single transition *replace*.

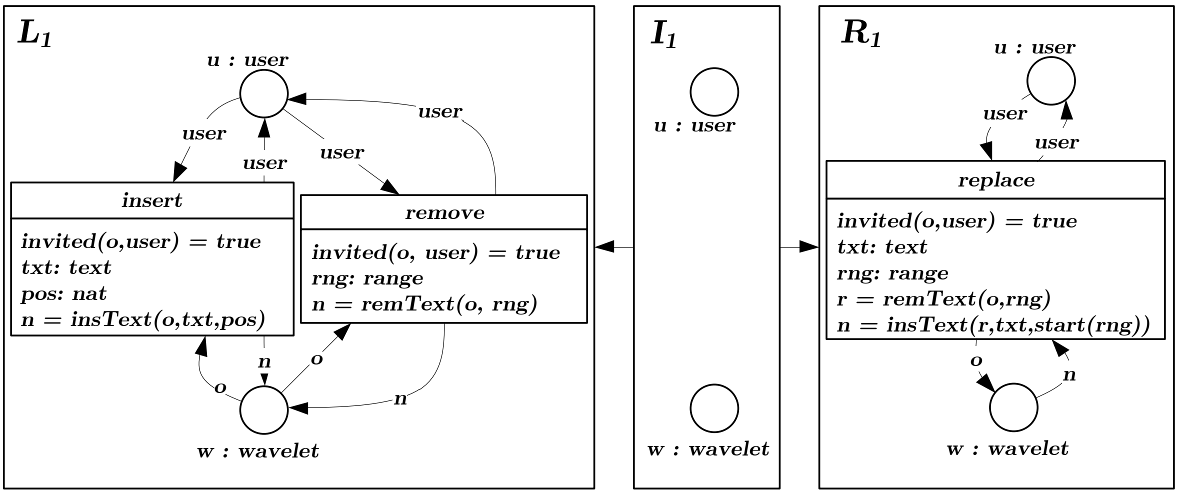


Figure 4: Production for AHL-nets

The gluing construction via pushouts in the category **AHLNets** of AHL-nets with fixed signature Σ and algebra A can be defined via the componentwise gluing of the sets as a pushout of the sets of places and transitions in the category **Sets**.

Definition 4.2 (Gluing of Sets)**.** Given sets A, B and C, and functions $f_1 : A \to B$, $f_2 : A \to C$. The gluing D of B and C along A (or more precisely along f_1 and f_2), written $D = B +_A C$, is defined as the quotient $D = (B \uplus C)/_\equiv$ where $\equiv$ is the smallest equivalence relation containing the relation

$$\sim \; = \; \{(f_1(a), f_2(a)) \mid a \in A\}.$$

This means that we transitively identify all those elements in $B \uplus C$ which are commonly mapped by the same interface element. Moreover, we obtain functions $g_1 : B \to D$ and $g_2 : C \to D$ with $g_1(b) = [b]_\equiv$ for all $b \in B$, and $g_2(c) = [c]_\equiv$ for all $c \in C$.

$$\begin{array}{ccc} A & \xrightarrow{f_1} & B \\ {\scriptstyle f_2}\downarrow & (\mathbf{PO}) & \downarrow{\scriptstyle g_1} \\ C & \xrightarrow[g_2]{} & D \end{array}$$

Fact 4.1 (Pushout of Sets)**.** *The diagram (PO) in Def. 4.2 is a* pushout diagram *in the category* **Sets***, i. e. (PO) commutes and has the following universal property: For all sets X and functions $h_1 : B \to X$, $h_2 : C \to X$ with $h_1 \circ f_1 = h_2 \circ f_2$ there exists a unique $h : D \to X$ with $h \circ g_1 = h_1$ and $h \circ g_2 = h_2$.*

Proof. See Fact 2.17 in [EEPT06]. □

Definition 4.3 (Gluing of AHL-Nets)**.** Given two AHL-net morphisms $f_1 : AN_0 \to AN_1$ and $f_2 : AN_0 \to AN_2$ the *gluing AN_3 of AN_1 and AN_2 along f_1 and f_2*, written $AN_3 = AN_1 +_{(AN_0,f_1,f_2)} AN_2$, with $AN_x = (\Sigma, P_x, T_x, pre_x, post_x, cond_x, type_x, A)$ for $x = 0, 1, 2, 3$ is constructed as follows:

- $T_3 = T_1 +_{T_0} T_2$ with $f'_{1,T}$ and $f'_{2,T}$ as pushout (2) of $f_{1,T}$ and $f_{2,T}$ in **Sets**.
- $P_3 = P_1 +_{P_0} P_2$ with $f'_{1,P}$ and $f'_{2,P}$ as pushout (3) of $f_{1,P}$ and $f_{2,P}$ in **Sets**
- $pre_3(t) = \begin{cases} f'^{\oplus}_{1,P} \circ pre_1(t_1) & \text{, if } f'_{1,T}(t_1) = t; \\ f'^{\oplus}_{2,P} \circ pre_2(t_2) & \text{, if } f'_{2,T}(t_2) = t. \end{cases}$
- $post_3(t) = \begin{cases} f'^{\oplus}_{1,P} \circ post_1(t_1) & \text{, if } f'_{1,T}(t_1) = t; \\ f'^{\oplus}_{2,P} \circ post_2(t_2) & \text{, if } f'_{2,T}(t_2) = t. \end{cases}$
- $cond_3(t) = \begin{cases} cond_1(t_1) & \text{, if } f'_{1,T}(t_1) = t; \\ cond_2(t_2) & \text{, if } f'_{2,T}(t_2) = t. \end{cases}$
- $type_3(p) = \begin{cases} type_1(p_1) & \text{, if } f'_{1,P}(p_1) = p; \\ type_2(p_2) & \text{, if } f'_{2,P}(p_2) = p. \end{cases}$
- $f'_1 = (f'_{1,P}, f'_{1,T})$ and $f'_2 = (f'_{2,P}, f'_{2,T})$.

$$\begin{array}{ccc} AN_0 & \xrightarrow{f_1} & AN_1 \\ {\scriptstyle f_2}\downarrow & (1) & \downarrow{\scriptstyle f'_1} \\ AN_2 & \xrightarrow[f'_2]{} & AN_3 \end{array} \qquad \begin{array}{ccc} T_0 & \xrightarrow{f_{1,T}} & T_1 \\ {\scriptstyle f_{2,T}}\downarrow & (2) & \downarrow{\scriptstyle f'_{1,T}} \\ T_2 & \xrightarrow[f'_{2,T}]{} & T_3 \end{array} \qquad \begin{array}{ccc} P_0 & \xrightarrow{f_{1,P}} & P_1 \\ {\scriptstyle f_{2,P}}\downarrow & (3) & \downarrow{\scriptstyle f'_{1,P}} \\ P_2 & \xrightarrow[f'_{2,P}]{} & P_3 \end{array}$$

Well-definedness. See Appendix A.2. □

Fact 4.2 (Pushout of AHL-Nets)**.** *The diagram (1) in Def. 4.3 is a* pushout diagram *in the category* **AHLNets***, i. e. (1) commutes and it has the following universal property: For all AHL-nets AN'_3 and AHL-morphisms $h_1 : AN_1 \to AN'_3$, $h_2 : AN_2 \to AN'_3$ with $h_1 \circ f_1 = h_2 \circ f_2$ there exists a unique AHL-morphism $h : AN_3 \to AN'_3$ such that $h \circ f'_1 = h_1$ and $h \circ f'_2 = h_2$.*

Proof-Idea. The pushouts (2) and (3) provide unique functions $h_P : P_3 \to P'_3$, $h_T : T_3 \to T'_3$ which together form an AHL-morphism $h = (h_P, h_T) : AN_3 \to AN'_3$ satisfying the universal property. For a detailed proof see Appendix A.3. □

Example 4.2 (Gluing and Transformation of Google Wave Platforms)**.** The gluing of two platforms GWP' and R_2 over an interface I_2 is shown in Fig. 5. The AHL-net GWP' is a modification of the Google Wave platform introduced in Section 2 in Fig. 1 where the ability to create a new wavelet is removed. The AHL-net R_2 is a smaller platform which does not provide any actions for the creation or editing of wavelets but only for the invitation and exclusion of users by other users. The net I_2 is a common subnet of both nets, containing the transition *invite* together with its pre and post domain.

The gluing of the nets GWP' and R_2 over a span of inclusions $GWP' \hookleftarrow I \hookrightarrow R_2$ leads to a net GWP_2 which is a combination of these two nets, providing actions for the creation and editing of wavelets as well as the invitation and exclusion of users.

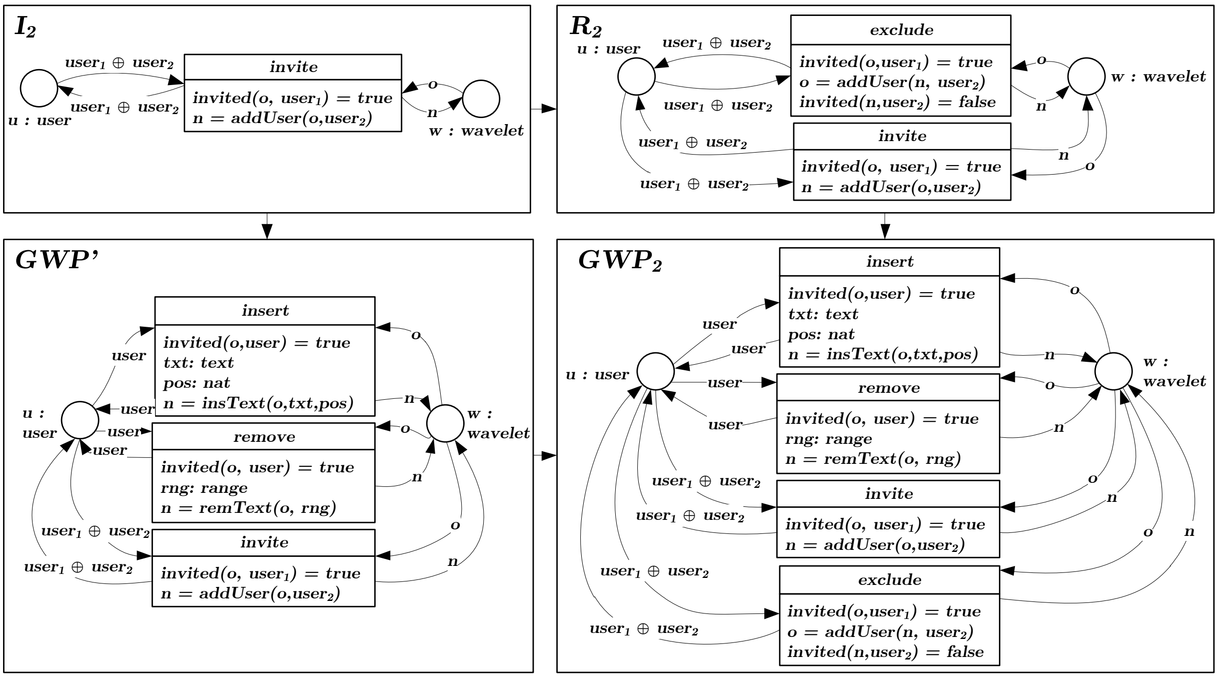


Figure 5: Gluing of AHL-nets

Note that there is only one transition *invite* in the net GWP_2 due to the fact that the corresponding transitions in the nets GWP' and R_2 are matched by the same transition in I_2. Otherwise, the gluing would result into two different copies of the *invite* transition.

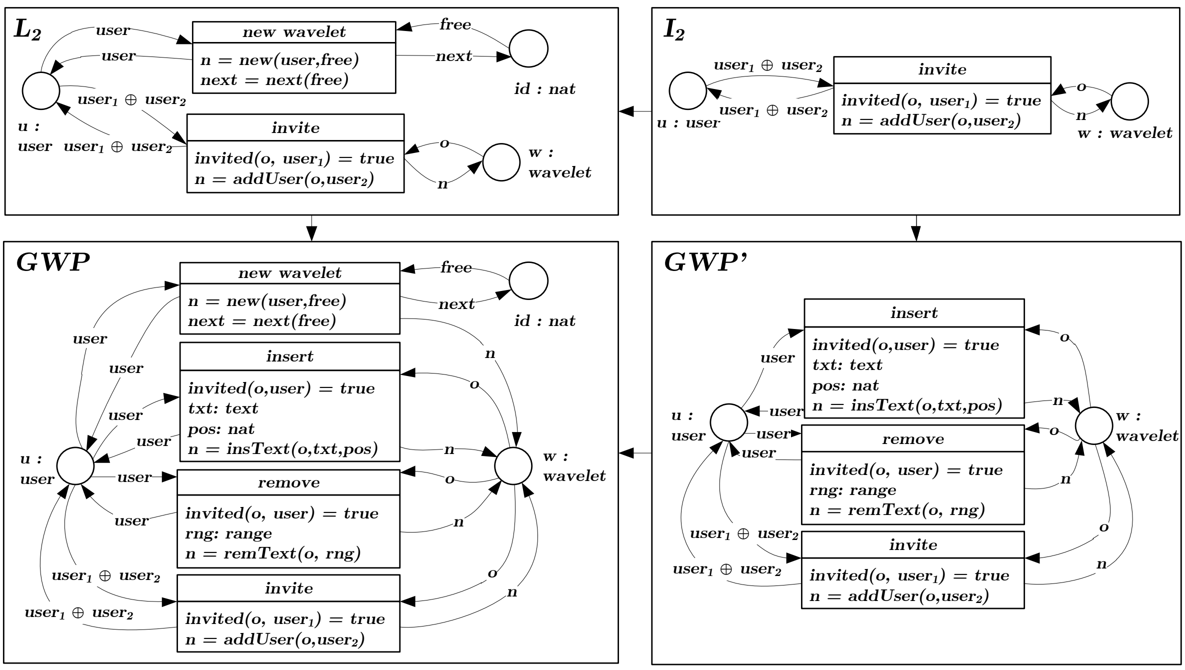


Figure 6: Restriction of AHL-nets

Consider a production for AHL-nets $p_2 : L_2 \stackrel{l_2}{\leftarrow} I_2 \stackrel{r_2}{\rightarrow} R_2$ where the net L_2 is shown in Fig. 6 and

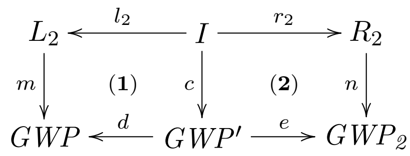
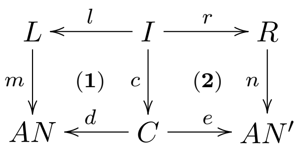


Figure 7: Direct Transformation $GWP \overset{p_2}{\Rightarrow} GWP_2$

the nets I_2 and R_2 are shown in Fig. 5. The production p_2 can be applied to the net GWP leading to the context AHL-net GWP' as shown in Fig. 6 and the result AHL-net GWP_2 as shown in Fig. 5, i. e. we have a direct tranformation $GWP \overset{p_2}{\Rightarrow} GWP_2$ (see Definition 4.4) with double pushout-diagram in Fig. 7, where (1) and (2) are pushouts.

Definition 4.4 (Direct Transformation of AHL-Nets)**.** Given a production $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ and a (match) morphism $m : L \rightarrow AN$ in **AHLNets**.

Then a *direct transformation* $AN \overset{(p,m)}{\Rightarrow} AN'$ in **AHLNets** is given by pushouts (1) and (2) in **AHLNets**. A transformation of AHL-nets is a sequence $AN_0 \overset{(p_1,m_1)}{\Rightarrow} AN_1 \cdots \overset{(p_n,m_n)}{\Rightarrow} AN_n$ of direct transformations, written $AN_0 \Rightarrow^* AN_n$.



Remark 4.1 (Modelling of Token-Game with Transformation)*.* For AHL-nets with individual tokens (see Remark 3.2) there is a similar definition for the rule-based direct transformation of AHL-nets with individual tokens (see [MGE+10]). It allows an alternative way to model the firing behaviour of AHL-nets by rule-based transformation. For every consistent transition assignment (t, asg) (see Def. 3.2) of an AHL-net with individual tokens ANI enabled under a token selection $S = (M, m, N, n)$ (see Remark 3.2) there is a corresponding *transition rule* $\varrho(t, S, asg)$ such that there is an equivalence between the firing of (t, asg) via S and the *canonical direct transformation* of ANI using the rule $\varrho(t, S, asg)$. For more details we refer to [MGE+10].

The following gluing condition is a necessary and sufficient condition for the existence of a direct transformation of AHL-nets. In order to satisfy the gluing condition by a production p under a match m some of the places and transitions in the AHL-net AN in the codomain of m must not be deleted by application of the production. The preimages of these elements in the left-hand side of the production are called identification points and dangling points.

The identification points are the preimages of places and transitions which are mapped non-injectively by the match m. The dangling points are the preimages of places which occur in the pre or post conditions of a transition which is matched, and therefore cannot be deleted by application of the production.

Definition 4.5 (Gluing Condition for AHL-Nets)**.** Given a production $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ for AHL-nets and an AHL-morphism $m : L \rightarrow AN$. We define the set of identification points[1]

$$\begin{aligned} IP \;=\; & \{x \in P_L \mid\; \exists\, x' \neq x : m_P(x) = m_P(x')\} \cup \\ & \{x \in T_L \mid\; \exists\, x' \neq x : m_T(x) = m_T(x')\} \end{aligned}$$

[1] i. e. all elements in L that are mapped non-injectively by m

the set of dangling points[2]

$$DP = \{p \in P_L \quad | \quad \exists\, t \in T_{AN} \setminus m_T(T_L), term \in T_\Sigma(X)_{type(p)} : \\ (term, m_P(p)) \leq pre_{AN}(t) \oplus post_{AN}(t)\}$$

and the set of gluing points[3]

$$GP = l_P(P_I) \cup l_T(T_I)$$

We say that p and m satisfy the gluing condition if $IP \cup DP \subseteq GP$.

$$\begin{array}{ccccc} L & \xleftarrow{l} & I & \xrightarrow{r} & R \\ {\scriptstyle m}\downarrow & & & & \\ AN & & & & \end{array}$$

Fact 4.3 (Transformation of AHL-Nets)**.** *Given a production for AHL-nets $p = (L \stackrel{l}{\leftarrow} I \stackrel{r}{\rightarrow} R)$ and a match $m : L \to AN$. The production p is applicable on match m, i. e. there exists a context AHL-net AN_0 in the diagram below, such that (1) is pushout, iff p and m satisfy the gluing condition in* **AHLNets**. *Then AN_0 is called* pushout complement *of l and m. Moreover, we obtain a unique AN' as pushout object of the pushout (2) in* **AHLNets**.

$$\begin{array}{ccccc} L & \xleftarrow{l} & I & \xrightarrow{r} & R \\ {\scriptstyle m}\downarrow & (1) & \downarrow{\scriptstyle c} & & \downarrow{\scriptstyle n} \\ AN & \underset{d}{\leftarrow} & AN_0 & \rightarrow & AN' \end{array}$$

If the AHL-net AN_0 exists it is unique up to isomorphism and can be constructed as follows:

- $P_{AN_0} = (P_{AN} \setminus m_P(P_L)) \cup m_P(l_P(P_I))$,
- $T_{AN_0} = (T_{AN} \setminus m_T(T_L)) \cup m_T(l_T(T_I))$,
- $pre_{AN_0} = pre_{AN}|_{T_{AN_0}}$, $post_{AN_0} = post_{AN}|_{T_{AN_0}}$, $cond_{AN_0} = cond_{AN}|_{T_{AN_0}}$ *and* $type_{AN_0} = type_{AN}|_{P_{AN_0}}$,
- *AN_0 has the same data part (Σ, A) as AN;*
- *$c_P(p) = m_P(l_P(p))$ for $p \in P_I$ and $c_T(t) = m_T(l_T(t))$ for $t \in T_I$, and*
- *d is an inclusion.*

Proof-Idea. AHL-nets can be seen as special cases of an AHL-nets with individual tokens where the set I of individual tokens is empty. Analogously every **AHLNets**-morphism can be seen as a special case of an **AHLINets**-morphism where the component for the individuals is the empty function. Therefore the proof of this fact works completely analogously to the proof of Fact 3.12 in [MGE+10].

For a detailed proof see Appendix A.4. □

[2] i. e. all places in L that would leave a dangling arc, if deleted
[3] i. e. all elements in L that have a preimage in I

5 Transformation of AHL-Occurrence Nets and AHL-Processes

In this section we extend our framework to the gluing and transformation of AHL-occurrence nets and processes. For this purpose we define productions for AHL-processes where the left hand and right hand side and the interface of the production are AHL-occurrence nets.

Definition 5.1 (Production for AHL-Processes)**.** A *production for AHL-processes* $p : L \stackrel{l}{\leftarrow} I \stackrel{r}{\rightarrow} R$ is a span of injective **AHLONets**-morphisms $l : I \to L$ and $r : I \to R$.

$$L \xleftarrow{\ l\ } I \xrightarrow{\ r\ } R$$

The following lemma states the fact that the gluing and the direct transformation of AHL-occurrence nets via pushout constructions can be computed in the category of AHL-nets because every pushout in **AHLONets** is also a pushout in **AHLNets**.

Lemma 5.1 (Pushout of AHL-Occurrence Nets)**.** *Given AHL-occurrence nets* I*,* K_1 *and* K_2 *and two AHL-net morphisms* $f : I \to K_1$ *and* $g : I \to K_2$*. If (1) is a pushout in* **AHLONets** *then (1) is also pushout in* **AHLNets**.

$$\begin{array}{ccccc} & & K_1 & & \\ & \nearrow^{f} & & \searrow^{f'} & \\ I & & (\mathbf{1}) & & K \\ & \searrow_{g} & & \nearrow_{g'} & \\ & & K_2 & & \end{array}$$

Proof Idea. Constructing the pushout of the given span in the category **AHLNets** we obtain a pushout object K' together with a unique induced morphism $k : K' \to K$. Then by Lemma 3.1 the morphism k implies that K' is an AHL-occurrence net leading to a unique morphism $k' : K \to K'$ by universal property of pushout (1). The morphisms k and k' can be shown to be inverse isomorphisms which by the uniqueness of pushouts implies that (1) is pushout in **AHLNets**. For a detailed proof see Appendix A.5. □

The gluing of AHL-nets may produce forward or backward conflicts as well as cycles or infinitely long chains in the causal relation. So for the gluing of two AHL-processes via pushout construction the processes have to be *composable* in order to obtain again an AHL-process as a result of the gluing. Composability of AHL-processes with respect to an interface means that the result of the gluing does not violate the process properties.

A span of injective **AHLONets**-morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$ induces a causal relation between the elements of the interface I. This relation consists of the causal relation between elements in K_1 and K_2 and additionally between those elements in both of the AHL-occurrence nets which is obtained by gluing over the interface.

Definition 5.2 (Induced Causal Relation)**.** Given three AHL-occurrence nets I, K_1 and K_2, and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$. The *induced causal relation* $<_{(i_1,i_2)}$ is defined as the transitive closure of the relation $\prec_{(i_1,i_2)}$ defined by

$$\prec_{(i_1,i_2)} = \{(x,y) \in (P_I \uplus T_I) \times (P_I \uplus T_I) \mid i_1(x) <_{K_1} i_1(y) \text{ or } i_2(x) <_{K_2} i_2(y)\}.$$

Definition 5.3 (Composability of AHL-Processes)**.** Given three AHL-occurrence nets I, K_1 and K_2, and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Then (K_1, K_2) *are composable w.r.t.* (I, i_1, i_2) if

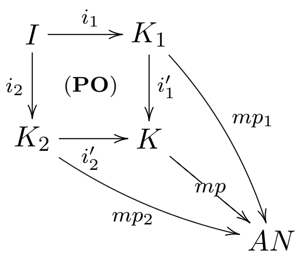
- $\forall\, x \in IN(I) : i_1(x) \notin IN(K_1) \Rightarrow i_2(x) \in IN(K_2)$,

- $\forall\, x \in OUT(I) : i_1(x) \notin OUT(K_1) \Rightarrow i_2(x) \in OUT(K_2)$, and
- the induced causal relation $<_{(i_1,i_2)}$ is a finitary strict partial order.

The composability of AHL-processes is a sufficient and necessary condition for the existence of the gluing of AHL-occurrence nets as pushout in the category **AHLONets**.

Fact 5.2 (Gluing of AHL-Processes)**.** *Given AHL-occurrence nets I, K_1, K_2 and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Then there exists a pushout (PO) in the category* **AHLONets** *(see Def. 3.3) iff (K_1, K_2) are composable w.r.t. (I, i_1, i_2). The AHL-occurrence net K is then called gluing of K_1 and K_2 along i_1 and i_2, written $K = K_1 +_{(I,i_1,i_2)} K_2$.*

In order to extend this gluing construction for AHL-processes in the category **Proc(AN)** *(see Def. 3.5) one additionally requires AHL-morphisms $mp_1 : K_1 \to AN$ and $mp_2 : K_2 \to AN$ with $mp_1 \circ i_1 = mp_2 \circ i_2$. The pushout (PO) in* **AHLONets** *then provides a unique morphism $mp : K \to AN$ such that (PO) is also a pushout in* **Proc(AN)***.*



Proof Idea. In order to show that pushout (PO) in **AHLNets** is also a pushout in the full subcategory **AHLONets** $\subseteq$ **AHLNets** it suffices to show that the pushout object K is an AHL-occurrence net. The conflict-freeness of K is ensured by the first two conditions of the required composability of K_1 and K_2 w.r.t. (I, i_1, i_2). The fact that the causal relation $<_K$ is a finitary strict partial order can be derived from the fact that the induced causal relation $<_{(i_1,i_2)}$ is a finitary strict partial order.

The other way around the pushout (PO) in **AHLONets** means that the pushout object K is an AHL-occurrence net and by Lemma 5.1 (PO) is also a pushout in **AHLNets**. Then by the fact that AHL-morphisms preserve pre and post conditions we can conclude from the conflict-freeness of K that the first two conditions of the composability of K_1 and K_2 are fulfilled. Moreover, the fact that $<_K$ is a finitary strict partial order implies that the same holds for the induced causal relation $<_{(i_1,i_2)}$.

For a detailed proof see Appendix A.6. □

Example 5.1 (Gluing of Waves)**.** The AHL-occurrence net *Wave* (Fig. 2 in Section 2) can be obtained by the gluing of the nets *Wave1* and *Wave2* over an interface I as shown in Fig. 8. Moreover, given AHL-morphisms $mp_1 : Wave_1 \to GWP$ and $mp_2 : Wave_2 \to GWP$ which maps every place and transition, respectively, to a place or transition with the same name without index we obtain a process morphism $mp : Wave \to GWP$.
Vice versa, we can consider the processes mp_1 and mp_2 as a decomposition of the process mp into two processes concerning the different wavelets in the wave.

Analogously to the induced causal relation we define a gluing relation for transformations which is induced by a production p and a match m. The gluing relation is a relation between the interface elements of p which consists of the causal relation between elements in the codomain of m which are preserved by application of p and the causal relation of the right hand side of the production, and additionally it consists of the causal relations which are obtained by gluing over the interface.

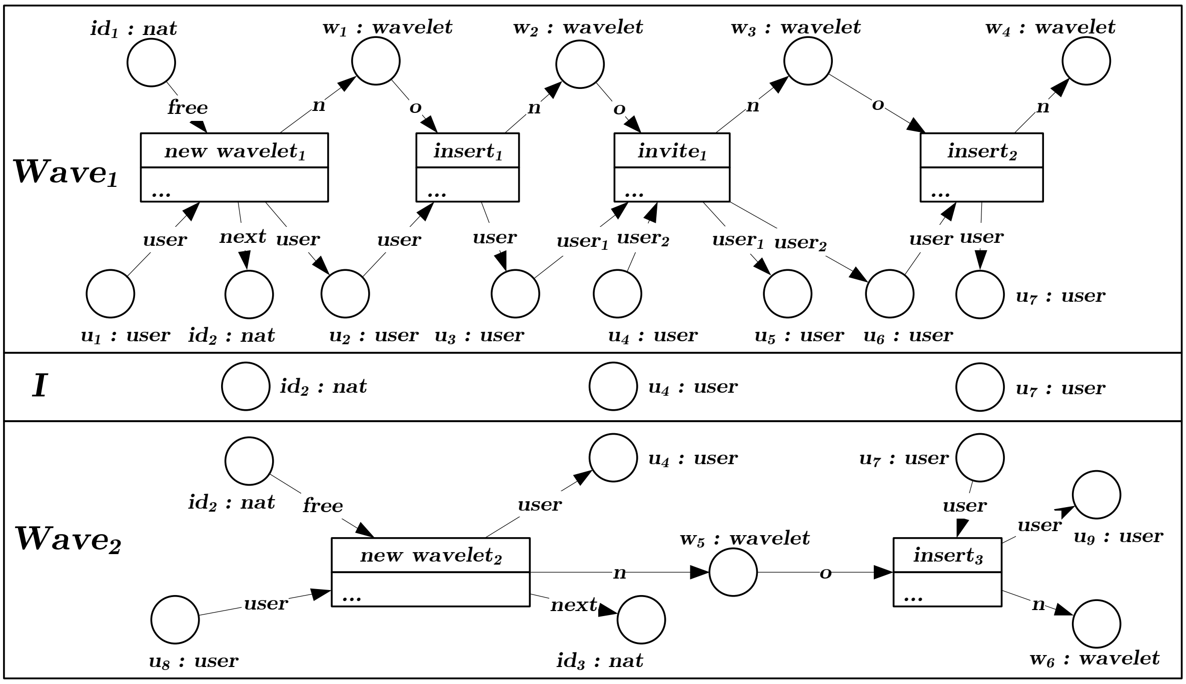


Figure 8: Gluing of Waves

Definition 5.4 (Gluing Relation for Transformations)**.** Given a production for AHL-processes $p : L \stackrel{l}{\leftarrow} I \stackrel{r}{\rightarrow} R$ and a match $m : L \rightarrow K$ we define the relations

$$\prec_{(K,m)} = \{(x, y) \in (P_K \times (T_K \setminus m_T(T_L))) \uplus ((T_K \setminus m_T(T_L)) \times P_K) \mid x \in \bullet y\}$$

and $<_{(K,m)}$ as the transitive closure of $\prec_{(K,m)}$. Furthermore we define

$$\prec_{(p,m)} = \{(x, y) \in (P_I \times T_I) \uplus (T_I \times P_I) \mid m \circ l(x) <_{(K,m)} m \circ l(y) \vee r(x) <_R r(y)\}$$

The transitive closure $<_{(p,m)}$ of $\prec_{(p,m)}$ is called *gluing relation* of production p under match m.

For the transformation of AHL-processes we define a *transformation condition* which is a necessary and sufficient condition that the direct transformation of an AHL-process exists. The satisfaction of the transformation condition by a production p and a match m requires that the gluing condition for AHL-nets (see Def. 4.5) is satsfied. Moreover, it requires that the gluing condition is finitary and irreflexive and that the application of the production does not produce any conflicts.

Definition 5.5 (Transformation Condition for AHL-Processes)**.** Given a production for AHL-processes $p : L \stackrel{l}{\leftarrow} I \stackrel{r}{\rightarrow} R$ and an AHL-occurrence net K. Then p satisfies the transformation condition under an injective (match) morphism $m : L \rightarrow K$ iff

- the gluing condition is satisfied,
- the gluing relation of p under m is a finitary strict partial order,
- for the sets of in and out places of the match

$$InP = \{x \in IN(I) \mid l(x) \in IN(L) \text{ and } m \circ l(x) \notin IN(K)\}, \text{ and}$$

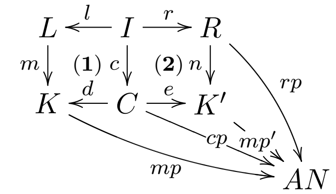
$$OutP = \{x \in OUT(I) \mid l(x) \in OUT(L) \text{ and } m \circ l(x) \notin OUT(K)\}$$

there is

$$r(InP) \subseteq IN(R) \text{ and } r(OutP) \subseteq OUT(R).$$

Theorem 5.3 (Direct Transformation of AHL-Processes)**.** *Given a production for AHL-processes* $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ *and an AHL-occurrence net* K *together with an injective morphism* $m : L \rightarrow K$*. Then the direct transformation of AHL-occurrence nets with pushouts (1) and (2) in* **AHLONets** *exists iff* p *satisfies the transformation condition for AHL-processes under* m*.*

In order to extend this construction for AHL-processes in the category **Proc**(**AN**) *one additionally requires AHL-morphisms* $mp : K \rightarrow AN$ *and* $rp : R \rightarrow AN$ *with* $mp \circ m \circ l = rp \circ r$*. Then the pushout (1) in* **AHLONets** *is a pushout of* $mp \circ m$ *and* $cp = mp \circ d$ *in* **Proc**(**AN**)*, and the pushout (2) in* **AHLONets** *provides a unique morphism* $mp' : K' \rightarrow AN$ *such that* mp' *is pushout of* cp *and* rp *in* **Proc**(**AN**) *according to Fact 5.2.*



Proof Idea. If the the transformation condition is satisfied we can construct pushout (1) in **AHLNets** leading to an AHL-occurrence net C by Lemma 3.1 and AHL-morphism $d : C \rightarrow K$. This means that (1) is also pushout in **AHLONets**. Moreover, the satisfaction of the transformation condition implies that (C, R) are composable w.r.t. (I, c, r) which by Fact 5.2 means that we can construct pushout (2) in **AHLONets**.

Vice versa, given pushouts (1) and (2) in **AHLONets** we have by Lemma 5.1 that (1) and (2) are also pushouts in **AHLNets** which means that production p with match m satisfies the gluing condition. The rest of the transformation condition can be obtained from the fact that (C, R) are composable w.r.t. (I, c, r) and the construction of pushout complement C.

For a detailed proof see Appendix A.7. □

Example 5.2 (Transformation of Waves)**.** Figure 9 shows a transformation of a Wave using a production for AHL-processes $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ and a match $m : L \rightarrow TwoWavelets$ where l, r and m are inclusions. The left-hand side of the production contains a *new wavelet₂* transition with its environment together with a *wavelet* place w_2 and a *user* place u_3. The *new wavelet₂* transition is then deleted by the production and instead the user u_5 which created the wavelet is then invited by user u_3 to the wavelet w_2 and the history of the deleted wavelet is appended to the wavelet w_2.

Note, that in order to apply the production the transition *new* $wavelet_2$ in the left-hand side of the production can only be matched to the transition *new* $wavelet_2$. A match $m' : L \rightarrow TwoWavelets$ which maps *new* $wavelet_2$ to *new* $wavelet_1$ does not satisfy the gluing condition for AHL-nets, because id_3 is a dangling point under match m' (see Def. 4.5).

Moreover, in order to obtain an AHL-occurrence net as result of the transformation, also the matching of the places is restricted. If we match the place w_2 to the place w_1 then the transformation would create a forward conflict at this place. If we match w_2 to w_4 then the transformation would create a cycle in the causal relation. Analogously a mapping of u_3 other than given by inclusion m would lead to the creation of a conflict or a cycle.

Now, consider process morphisms $mp_1 : TwoWavelets \rightarrow AN$ and $mp_2 : R \rightarrow AN$ which map the places and transitions to the corresponding elements in AN without index. Then from the transformation $TwoWavelets \overset{p,m}{\Rightarrow} OneWavelet$ in **AHLONets** we obtain a process morphism $mp_3 : OneWavelet \rightarrow AN$ which maps the elements in the same way as mp_1 and mp_2.

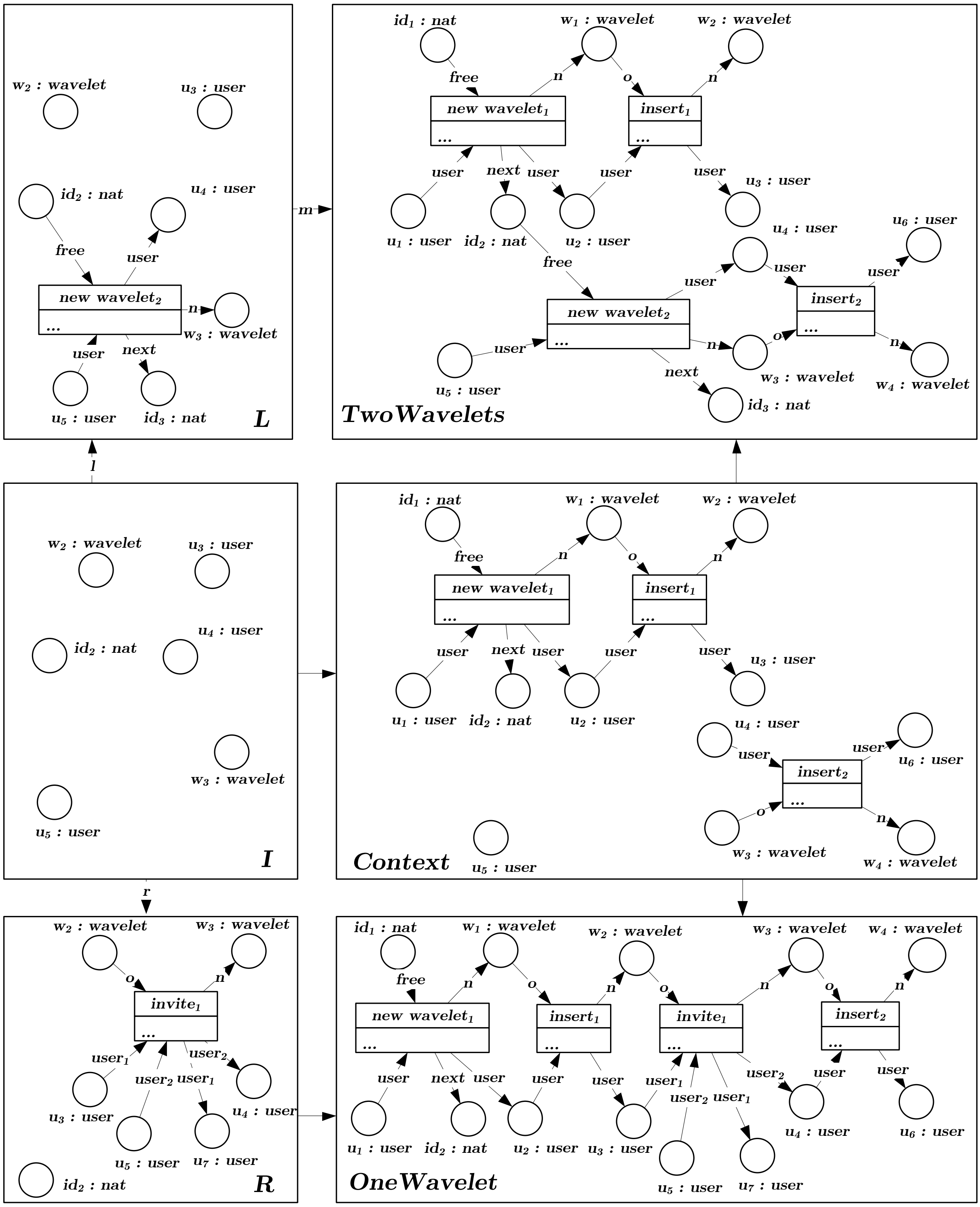


Figure 9: Transformation of a Wave

6 Compatibility of AHL-Net Transformations

Due to the fact that Google Wave is open source it is possible that different developers perform their own evolutions outgoing from one platform. It is an interesting aspect to analyse whether such different evolutions are compatible with each other in the sense that each one of the evolutions can be also applied to the result of the respective other one leading to the same result.

Example 6.1 (Compatible Platform Evolutions)**.** Consider again the production for AHL-nets p_1 in Example 4.1 and the production p_2 in Example 4.2. Both of the productions are applicable to the Google Wave platform *GWP* leading to AHL-nets GWP_1 and GWP_2 as shown in Fig 10.
The two transformations $GWP \stackrel{p_1}{\Rightarrow} GWP_1$ and $GWP \stackrel{p_2}{\Rightarrow} GWP_2$ are compatible in the sense that each of the productions is applicable to the result of the transformation with the respective other production, leading to the same result GWP_3 as shown in Fig. 10.
Note that this would not be possible if one of the productions would delete something that is needed for the applicability of the other production, e. g. if p_1 would delete the transition *invite* or the place w.

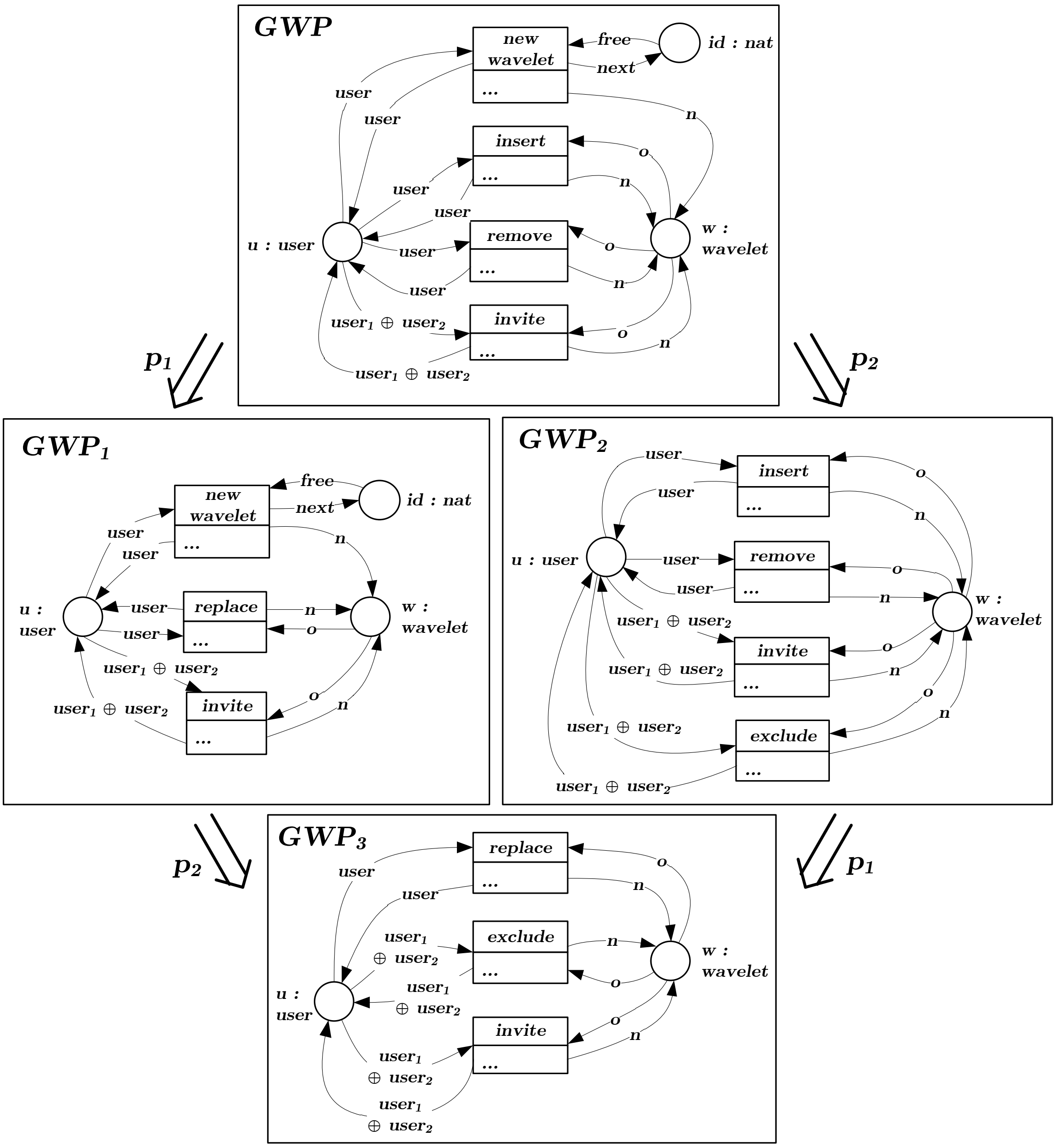
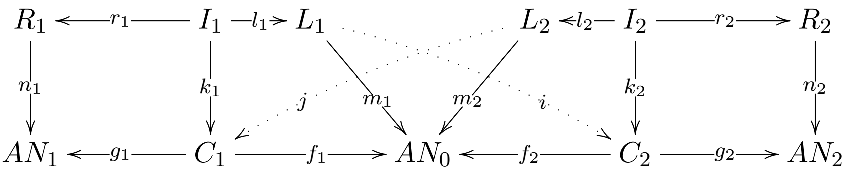


Figure 10: Compatible Platform Evolutions

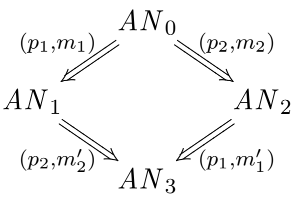
In the following we show that direct transformations can be applied in any order (see Theorem 6.1), provided that they are parallel independent in the following sense.

Definition 6.1 (Parallel Independence of AHL-Net Transformations)**.** Two direct transformations of AHL-nets $AN_0 \overset{(p_1,m_1)}{\Longrightarrow} AN_1$ and $AN_0 \overset{(p_2,m_2)}{\Longrightarrow} AN_2$ are called *parallel independent* if there exist morphisms $i : L_1 \to C_2$ and $j : L_2 \to C_1$ such that $f_2 \circ j = m_1$ and $f_1 \circ j = m_2$.



Remark 6.1.
1. **(Characterization of Parallel Independence)** Parallel independence is equivalent to the fact that the matches only overlap in gluing points, i. e. $m_1(L_1) \cap m_2(L_2) \subseteq l_1(m_1(I_1)) \cap l_2(m_2(I_2))$.
2. **(Sequential Independence)** Analogously to the parallel independence of AHL-nets it is possible to define a sequential independence by defining that $AN_0 \overset{p_1}{\Rightarrow} AN_1 \overset{p_2}{\Rightarrow} AN_2$ are sequentially independent iff $AN_0 \overset{p_1^{-1}}{\Leftarrow} AN_1 \overset{p_2}{\Rightarrow} AN_2$ are parallel independent.

Theorem 6.1 (Local Church-Rosser Theorem for AHL-Net Transformations)**.** *Given two parallel independent direct transformations* $AN_0 \overset{(p_1,m_1)}{\Longrightarrow} AN_1$ *and* $AN_0 \overset{(p_2,m_2)}{\Longrightarrow} AN_2$ *then there is an AHL-net* AN_3 *together with direct transformations* $AN_1 \overset{(p_2,m_2')}{\Longrightarrow} AN_3$ *and* $AN_2 \overset{(p_1,m_1')}{\Longrightarrow} AN_3$*. Moreover, the following transformation sequences are sequential independent, such that we can apply first production* p_1 *and then* p_2 *and vice versa.*



Proof. The Local Church-Rosser Theorem has originally been shown for graph transformation systems in [ER76] and it is shown in [EEPT06] in the categorical framework of "high-level replacement systems" based on weak adhesive HLR categories. The category $(\mathbf{AHLNets}, \mathcal{M})$ with class $\mathcal{M}$ of all injective AHL-net morphisms is shown to be a weak adhesive HLR category in [EEPT06] Fact 4.25, such that the Local Church-Rosser Theorem is also valid for AHL-net transformations. □

Remark 6.2 (Weak Adhesive HLR-Category and Van Kampen-Property). Roughly spoken a weak adhesive HLR-category $(\mathbf{C}, \mathcal{M})$ is a category $\mathbf{C}$ with suitable class $\mathcal{M}$ of monomorphisms, such that

1. $\mathbf{C}$ has pushouts and pullbacks along $\mathcal{M}$-morphisms,
2. $\mathcal{M}$ is closed under pushouts and pullbacks, and
3. pushouts along $\mathcal{M}$ are weak Van Kampen (VK) squares, i. e. they are compatible with pullbacks along $\mathcal{M}$ and vice versa (see [EEPT06] for more details).

In our weak adhesive HLR-category $(\mathbf{AHLNets}, \mathcal{M})$ pushouts and pullbacks are defined in Definition 3.6 and Definition 4.3, where for simplicity inclusions are used instead of injective AHL-morphisms $m \in \mathcal{M}$. The VK-property will be used in the proof of Theorem 7.1 (Amalgamation of Processes).

Note that Example 6.1 is an example for Theorem 6.1, where the AHL-nets model different Google Wave platforms. A similar evolution result is also interesting in the cases of waves, but this case is technically more difficult. As pointed out in Fact 5.2 we need additional conditions for the construction of pushouts of AHL-processes. This implies that the category $\mathbf{Proc}(\mathbf{AN})$ of AHL-processes over AN does not define a suitable weak adhesive HLR category, such that the general Local Church-Rosser Theorem cannot be applied in the case of processes.

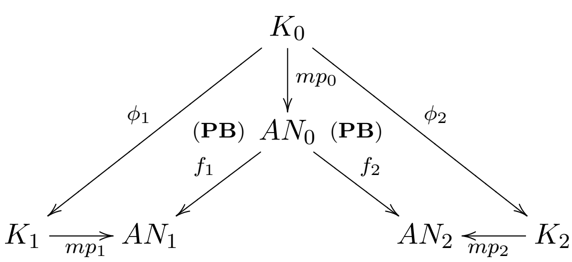
7 Amalgamation of AHL-Processes and Compositional Process Semantics

Google Wave is open source and thus there is not only one possible Wave platform but many different ones. Due to the Wave federation protocol it is possible that users on different servers can communicate within one wave. For the history of a wave this means that different parts of the history correspond to different platforms. Considering the process model of a wave this means that in this case the process is composed of different processes with respect to different AHL-nets, whereas the complete wave model is a process with respect to the gluing of these AHL-nets.

Example 7.1 (Amalgamation of Waves)**.** Consider the Google Wave platforms GWP_B and GWP_C with waves $Wave_B$ and $Wave_C$, respectively, in Fig. 11. The platform GWP_A is a subnet of GWP_B and GWP_C containing the features that these two platforms have in common. We can obtain a wave $Wave_A$ of the platform GWP_A by the restriction of $Wave_B$ via f_1 as well as by restriction of $Wave_C$ via f_2. The wave consists of the activities that appear in both of the waves $Wave_B$ and $Wave_C$.
By the gluing of the platforms GWP_B and GWP_C over GWP_A we obtain a platform GWP_D consisting of the whole set of features from GWP_B and GWP_C. Due to the fact that the two waves $Wave_B$ and $Wave_C$ are composable w.r.t. $Wave_A$ they can be glued together leading to $Wave_D$, called amalgamation of $Wave_B$ and $Wave_C$ along $Wave_A$. $Wave_D$ is a wave of the platform GWP_D capturing all activities. Restricting the wave $Wave_D$ via the morphism g_1 respectively g_2 we obtain again the wave $Wave_B$ respectively $Wave_C$.
Note that this is only possible under certain condition, e. g. consider the case that the AHL-occurrence net $Wave_C$ is modified to a net $Wave'_C$ which contains no place u_1. Then by the restriction of $Wave'_C$ via f_2 we obtain a net $Wave'_A$ containing no corresponding place u_1 which means that $Wave'_A$ is not the restriction of $Wave_B$ via f_1. Anyway, since $Wave'_A$ is a subnet of $Wave_A$ there is also a morphism $\phi'_1 : Wave'_A \to Wave_B$ and $Wave_B$, $Wave_C$ can be glued together over the alternative interface leading again to the same AHL-occurrence net $Wave_D$, but $Wave'_C$ is not the restriction of $Wave_D$ via g_2. Moreover, there are also other cases where we do not even have a suitable interface for the gluing of the AHL-occurrence nets $Wave_B$ and $Wave_C$. In the following we discuss the general conditions for an amalgamation of AHL-processes.

Based on the restriction of AHL-processes (see Fact 3.2) we can define a suitable condition under which we can continue the composition of AHL-nets via a span of injective AHL-morphisms $f_1 : AN_0 \to AN_1$ and $f_2 : AN_0 \to AN_2$ to a composition of their processes. Two processes mp_1 and mp_2 of AN_1 and AN_2, respectively, "agree" on the net AN_0 if we can construct a common restriction of the processes leading to a process mp_0 of AN_0 which can be used as a composition interface for mp_1 and mp_2.

Definition 7.1 (Agreement of AHL-Processes)**.** Given two AHL-processes $mp_1 : K_1 \to AN_1$ and $mp_2 : K_2 \to AN_2$ and two injective AHL-morphisms $f_1 : AN_0 \to AN_1$, $f_2 : AN_0 \to AN_2$. The processes mp_1 and mp_2 agree on mp_0 if there exist restrictions (mp_0, ϕ_i) of mp_i along f_i for $i \in \{1, 2\}$ such that for $mp_0 : K_0 \to AN_0$ the AHL-occurrence nets K_1 and K_2 are composable via ϕ_1, ϕ_2, i. e. the gluing of K_1 and K_2 along K_0 exists and leads to an occurrence net K_3.
(mp_0, ϕ_1) and (mp_0, ϕ_2) are called agreement restrictions for mp_1 and mp_2.



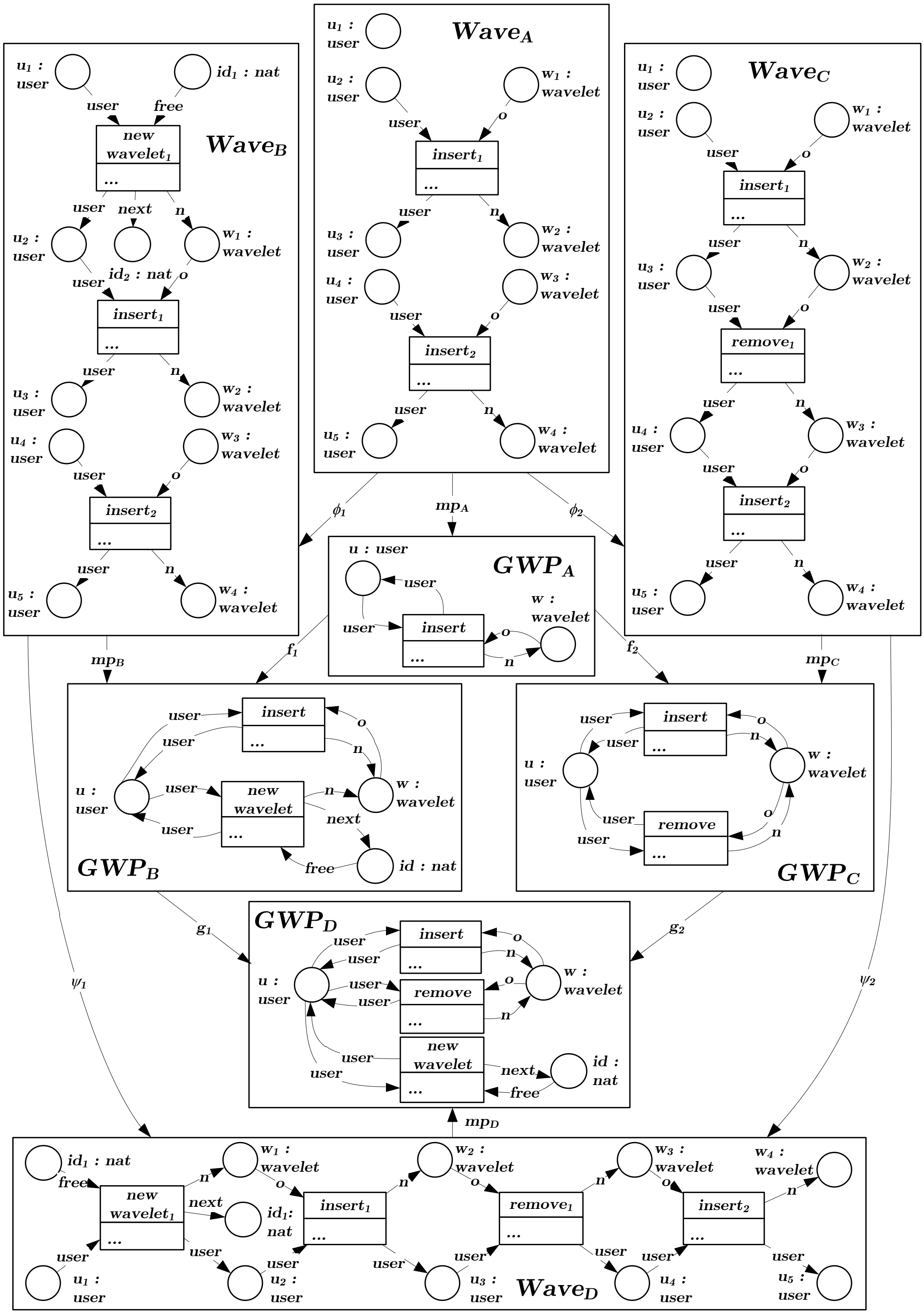


Figure 11: Amalgamation of Waves

If two processes $mp_1 : K_1 \to AN_1$ and $mp_2 : K_2 \to AN_2$ agree then they can be amalgamated. This means that they are composed to a process $mp_3 : K_3 \to AN_3$ of AN_3, which is the composition of AN_1 and AN_2, such that mp_1 and mp_2 are restrictions of mp_3.

Definition 7.2 (Amalgamation of AHL-Processes)**.** Let AN_3 be the gluing (pushout) of AHL- nets AN_1 and AN_2 over injective morphisms f_1 and f_2 as shown in (PO) of Figure 12, and $mp_i : K_i \to AN_i$ be AHL-processes for $i \in \{0, 1, 2, 3\}$ s.t. mp_1 and mp_2 agree on mp_0, i. e. (1) and (2) are pullbacks s.t. K_1 and K_2 are composable via ϕ_1, ϕ_2. This means especially that the outer square in Fig. 12 is a pushout of AHL-nets. Then mp_3 is called amalgamation of mp_1 and mp_2 along mp_0, written $mp_3 = mp_1 +_{mp_0} mp_2$, if there exist restrictions (mp_1, ψ_1) and (mp_2, ψ_2) of mp_3 along g_1 and g_2 in (3) and (4).

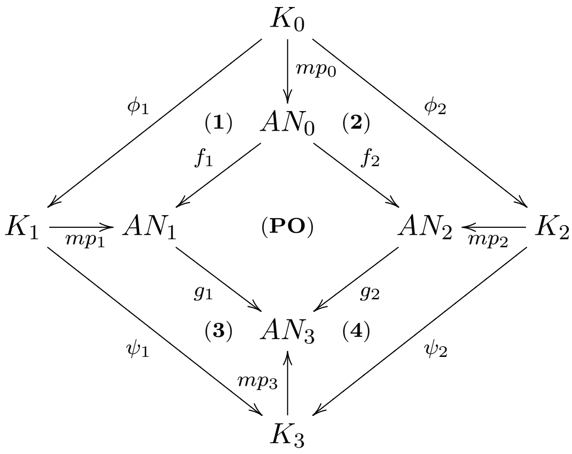


Figure 12: Amalgamation of AHL-Processes

The results of amalgamation composition and decomposition constructions are unique up to isomorphism. In order to capture the bijective correspondence of these constructions we define isomorphism classes of AHL-processes and spans of AHL-processes analogously to isomorphism classes of open net processes and spans of these processes in [BCEH01].

An isomorphism between processes $mp : K \to AN$ and $mp' : K' \to AN$ of an AHL-net AN is an isomorphism $iso : K \to K'$ in the category **AHLNets** which is also a morphism in **Proc**(**AN**), i. e. diagram (1) in Figure 13 commutes. We denote the isomorphism class of a process mp as the set $[mp] = \{mp' \mid mp' \cong mp\}$ of all processes which are isomorphic to mp.

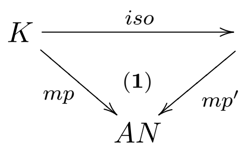


Figure 13: Isomorphism of Processes

An isomorphism of spans of processes $(mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2) \cong (mp'_1 \overset{\phi'_1}{\leftarrow} mp'_0 \overset{\phi'_2}{\rightarrow} mp'_2)$ means that there are process isomorphisms $iso_i : mp_i \to mp'_i$ such that the diagram in Figure 14 commutes.

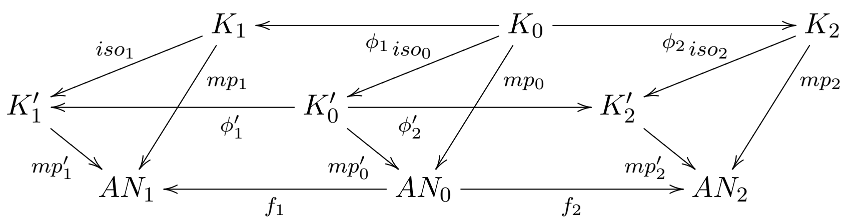


Figure 14: Isomorphism of spans of processes

Definition 7.3 (Sets of Isomorphism Classes)**.** The set of all isomorphism classes of processes of a given AHL-net AN is defined as

$$Proc(AN) = \{[mp] \mid mp : K \to AN \text{ is a process }\}$$

The set of all isomorphism classes of spans of agreeing AHL-processes with respect to a given span of AHL-morphisms is defined as

$$Proc(AN_1 \overset{f_1}{\leftarrow} AN_0 \overset{f_2}{\rightarrow} AN_2) =$$

$$\{\left[mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2\right] \mid \phi_1, \phi_2 \text{ are agreement projections of } mp_1, mp_2 \text{ along } f_1, f_2\}$$

Theorem 7.1 (Amalgamation Theorem for AHL-Processes)**.** *Given the gluing (pushout) $AN_3 = AN_1 +_{AN_0} AN_2$ of AHL-nets in (PO) of Fig. 12 with injective f_1, f_2 then we have:*

1. ***Composition Construction.*** *Let $mp_i : K_i \to AN_i$ be AHL-processes for $i \in \{0, 1, 2\}$ such that mp_1 and mp_2 agree on mp_0, then the amalgamation $mp_3 = mp_1 +_{mp_0} mp_2$ exists and is an AHL-process $mp_3 : K_3 \to AN_3$.*

2. ***Decomposition Construction.*** *Let $mp_3 : K_3 \to AN_3$ be an AHL-process and let mp_1, mp_2 be restrictions of mp_3 along g_1 resp. g_2, and mp_0 restriction of mp_1 along f_1. Then mp_3 can be represented as amalgamation $mp_3 = mp_1 +_{mp_0} mp_2$.*

3. ***Bijective Correspondence.*** *There are composition and decomposition functions*

   $$Comp : Proc(AN_1 \overset{f_1}{\leftarrow} AN_0 \overset{f_2}{\rightarrow} AN_2) \to Proc(AN)$$

   and

   $$Decomp : Proc(AN) \to Proc(AN_1 \overset{f_1}{\leftarrow} AN_0 \overset{f_2}{\rightarrow} AN_2)$$

   establishing a bijective correspondence between $Proc(AN)$ and $Proc(AN_1 \overset{f_1}{\leftarrow} AN_0 \overset{f_2}{\rightarrow} AN_2)$.

Proof Idea. 1. Agreement of mp_1 and mp_2 on mp_0 impies that mp_0 exists in Fig. 12 s.t. (1) and (2) are pullbacks and the outer diagram can be constructed as pushout of AHL-nets leading to an occurrence net K_3. The universal pushout property implies a unique $mp_3 : K_3 \to AN_3$ s.t. (3) and (4) commute. Finally, the VK-property for the weak adhesive HLR category $(\mathbf{AHLNets}, \mathcal{M})$ implies that (3) and (4) are pullbacks s.t. mp_1 and mp_2 become restrictions of mp_3 along g_1 resp. g_2 leading to the amalgamation $mp_3 = mp_1 +_{mp_0} mp_2$.

2. In this case we have given in Fig. 12 the inner pushout (PO) and $mp_3 : K_3 \to AN_3$. Now mp_1, mp_2, mp_0 are constructed as pullbacks in (3), (4) resp. (1), and (2) can be shown to become pullback s.t. K_1 and K_2 are composable via ϕ_1, ϕ_2. Finally, the VK-property in Fig. 12 implies that the outer diagram is pushout, s.t. $mp_3 = mp_1 +_{mp_0} mp_2$ becomes the amalgamation of mp_1 and mp_2 along mp_0.

3. Follows from uniqueness (up to isomorphism) of pushout and pullback constructions.

For a detailed proof see Appendix A.8. □

Theorem 7.1 implies that we have a compositional process semantics of AHL-nets in the following sense.

Corollary 7.2 (Compositional Process Semantics of AHL-Nets)**.** *Given AHL-nets AN_i $(i = 0, 1, 2, 3)$ with $AN_3 = AN_1 +_{AN_0} AN_2$ gluing (pushout) object in (PO) of Fig. 12. Then each process $mp_3 : K_3 \to AN_3$ of AN_3 is uniquely (up to isomorphism) represented by a pair of processes ($mp_1 : K_1 \to AN_1$, $mp_2 : K_2 \to AN_2$) of AN_1 resp. AN_2, which agree on mp_0 obtained as common restriction of mp_1 and mp_2. This means that the process semantics of AN_3, defined by all processes $mp_3 : K_3 \to AN_3$ over AN_3, is completely determined by the process semantics of the components AN_1 and AN_2 with shared AN_0.*

8 Conclusion

In this paper we have presented a comprehensive introduction to algebraic high-level (AHL) nets, AHL-processes and rule-based transformations of AHL-nets and their processes as integrated framework for modeling communication based systems and communication platforms. In this paper we have chosen as case study and running example Google Wave, while high-level net transformations have been applied already successfully to Skype [MEE+10]. A process algebraic modeling of the main communication algorithm from a technical point of view is presented in [Yon10]. In future work we will study how to combine these different views and to model other important features of communication platforms like Google Wave. This includes also an extension of the modeling framework to the gluing and transformation of AHL-processes with instantiations. Moreover we want to consider interesting consistency and security requirements and study how they can be satisfied in our model based on the current integrated framework, or how to extend the model and/or the framework respectively.

A Detailed Proofs

A.1 Proof of Lemma 3.1 (AHL-Morphisms Reflect AHL-Occurrence Nets)

Given an AHL-morphism $f : K_1 \to K_2$. If K_2 is an AHL-occurrence net then also K_1.

Proof. Given AHL-morphism $f : K_1 \to K_2$ with AHL-occurrence net K_2. In order to show that K_1 is an AHL-occurrence net we have to show that it is unary, there are no forward or backward conflicts and the causal relation $<_{K_1}$ is a finitary strict partial order.

Unarity. Let us assume that K_1 is not unary, i. e. there are $p \in P_{K_1}$, $t \in T_{K_1}$ with

$$(term_1, p) \oplus (term_2, p) \leq pre_{K_1}(t) \text{ or } (term_1, p) \oplus (term_2, p) \leq post_{K_1}(t)$$

Let $(term_1, p) \oplus (term_2, p) \leq pre_{K_1}(t)$.
Since AHL-morphisms preserve pre conditions there is

$$(id_{T_{OP}(X)} \otimes f_P)^{\oplus} \circ pre_{K_1}(t) = pre_{K_2}(f_T(t))$$

and hence

$$\begin{aligned}(term_1, f_P(p)) \oplus (term_2, f_P(p)) &= (id_{T_{OP}(X)} \otimes f_P)^{\oplus}((term_1, p) \oplus (term_2, p)) \\ &\leq pre_{K_2}(f_T(t))\end{aligned}$$

This implies that K_2 is not unary, contradicting the fact that K_2 is an AHL-occurrence net. The case that $(term_1, p) \oplus (term_2, p) \leq post_{K_1}(t)$ works analogously. Hence K_1 is unary.

No forward conflict. Let us assume that K_1 has a forward conflict, i. e. there is $p \in P_{K_1}$, $t_1 \neq t_2 \in T_{K_1}$ with $p \in \bullet t_1 \cap \bullet t_2$. This means that there are $term_1, term_2 \in T_{OP}(X)_{type(p)}$ such that

$$(term_1, p) \leq pre_{K_1}(t_1) \text{ and } (term_2, p) \leq pre_{K_1}(t_2)$$

and since AHL-morphisms preserve pre and post conditions we obtain

$$\begin{aligned}(term_1, f_P(p)) &= (id_{T_{OP}(X)} \otimes f_P)^{\oplus}(term_1, p) \\ &\leq pre_{K_2}(f_T(t_1))\end{aligned}$$

and

$$\begin{aligned}(term_2, f_P(p)) &= (id_{T_{OP}(X)} \otimes f_P)^{\oplus}(term_2, p) \\ &\leq pre_{K_2}(f_T(t_2))\end{aligned}$$

In the case that $f_T(t_1) \neq f_T(t_2)$ the fact that $f_P(p) \in \bullet f_T(t_1) \cap \bullet f_T(t_2)$ means that K_2 has a forward conflict, contradicting the fact that K_2 is an AHL-occurrence net. So let us consider the fact that $f_T(t_1) = t = f_T(t_2)$. Then we have

$$(term_1, f_P(p)) \oplus (term_2, f_P(p)) \leq t$$

which contradicts the fact that K_2 is unary. Hence K_1 has no forward conflict.

No backward conflict. The proof for this case works analogously to the one for forward conflicts because AHL-morphisms preserve post as well as pre conditions and K_2 has no backward conflicts.

Finitary strict partial order. We have to show that $<_{K_1}$ is finitary and irreflexive.

Finitariness. Let us assume that $<_{K_1}$ is not finitary. Then there is an element $x \in P_{K_1} \uplus T_{K_1}$ with an infinite number of predecessors. Let

$$S = \{y \in P_{K_1} \uplus T_{K_1} \mid y <_{K_1} x\}$$

be the infinite set of predecessors of x. Since AHL-morphisms preserve pre and post conditions, $a <_{K_1} b$ implies $f(a) <_{K_2} f(b)$. This means that there is an infinite set

$$S' = \{f(y) \mid y \in S\}$$

where for every $f(y) \in S'$ there is

$$f(y) <_{K_2} f(x)$$

This means that $f(x)$ has an infinite number of predecessors implying that $<_{K_2}$ is not finitary. This contradicts the fact that K_2 is an AHL-occurrence net and hence $<_{K_1}$ is finitary.

Irreflexivity. Let us assume that $<_{K_1}$ is not irreflexive, i.e. there exists a cycle $x <_{K_1} x$. This implies $f(x) <_{K_2} f(x)$ contradicting the fact that $<_{K_2}$ is irreflexive and hence $<_{K_1}$ is irreflexive.

□

A.2 Well-definedness of Definition 4.3 (Gluing of AHL-Nets)

Given two AHL-net morphisms $f_1 : AN_0 \to AN_1$ and $f_2 : AN_0 \to AN_2$ the *gluing* AN_3 *of* AN_1 *and* AN_2 *along* f_1 *and* f_2, written $AN_3 = AN_1 +_{(AN_0, f_1, f_2)} AN_2$, with

$$AN_x = (\Sigma, P_x, T_x, pre_x, post_x, cond_x, type_x, A)$$

for $x = 0, 1, 2, 3$ is constructed as follows:

- $T_3 = T_1 +_{T_0} T_2$ with $f'_{1,T}$ and $f'_{2,T}$ as pushout (2) of $f_{1,T}$ and $f_{2,T}$ in **Sets**.
- $P_3 = P_1 +_{P_0} P_2$ with $f'_{1,P}$ and $f'_{2,P}$ as pushout (3) of $f_{1,P}$ and $f_{2,P}$ in **Sets**
- $pre_3(t) = \begin{cases} f'^{\oplus}_{1,P} \circ pre_1(t_1) & \text{, if } f'_{1,T}(t_1) = t; \\ f'^{\oplus}_{2,P} \circ pre_2(t_2) & \text{, if } f'_{2,T}(t_2) = t. \end{cases}$
- $post_3(t) = \begin{cases} f'^{\oplus}_{1,P} \circ post_1(t_1) & \text{, if } f'_{1,T}(t_1) = t; \\ f'^{\oplus}_{2,P} \circ post_2(t_2) & \text{, if } f'_{2,T}(t_2) = t. \end{cases}$

- $cond_3(t) = \begin{cases} cond_1(t_1) & \text{, if } f'_{1,T}(t_1) = t; \\ cond_2(t_2) & \text{, if } f'_{2,T}(t_2) = t. \end{cases}$

- $type_3(p) = \begin{cases} type_1(p_1) & \text{, if } f'_{1,P}(p_1) = p; \\ type_2(p_2) & \text{, if } f'_{2,P}(p_2) = p. \end{cases}$

- $f'_1 = (f'_{1,P}, f'_{1,T})$ and $f'_2 = (f'_{2,P}, f'_{2,T})$.

$$\begin{array}{ccc} AN_0 & \xrightarrow{f_1} & AN_1 \\ {\scriptstyle f_2}\downarrow & (\mathbf{1}) & \downarrow{\scriptstyle f'_1} \\ AN_2 & \xrightarrow[f'_2]{} & AN_3 \end{array} \qquad \begin{array}{ccc} T_0 & \xrightarrow{f_{1,T}} & T_1 \\ {\scriptstyle f_{2,T}}\downarrow & (\mathbf{2}) & \downarrow{\scriptstyle f'_{1,T}} \\ T_2 & \xrightarrow[f'_{2,T}]{} & T_3 \end{array} \qquad \begin{array}{ccc} P_0 & \xrightarrow{f_{1,P}} & P_1 \\ {\scriptstyle f_{2,P}}\downarrow & (\mathbf{3}) & \downarrow{\scriptstyle f'_{1,P}} \\ P_2 & \xrightarrow[f'_{2,P}]{} & P_3 \end{array}$$

Well-definedness.

Well-definedness of AHL-net AN_3. We have to show that the definition of the pre, post and firing conditions together with the type function form an AHL-net. In the definition of each of the functions for every $t \in T_3$ or $p \in P_3$, respectively, at least one of the cases occur due to the definition of the gluing T_3.
Let us consider a transition $t \in T_3$ such that there are $t_1 \in T_1$, $t_2 \in T_2$ such that $f'_{1,T}(t_1) = t$ and $f'_{2,T}(t_2) = t$. Then by the transitive closure of $\equiv$ (see Definition 4.2) there exist $a_0, \ldots, a_n \in T_0$ with $f_{1,T}(a_0) = t_1$, $f_{2,T}(a_0) = f_{2,T}(a_1)$, $f_{1,T}(a_1) = f_{1,T}(a_2)$,..., $f_{2,T}(a_{n-1}) = f_{2,T}(a_n) = t_2$.
By the fact that f_1 and f_2 are AHL-morphisms which preserve pre conditions we have

$$f^{\oplus}_{1,P}(pre_0(a_0)) = pre_1(t_1),$$

$$f^{\oplus}_{2,P}(pre_0(a_0)) = pre_2(f_{2,T}(a_0)) = pre_2(f_{2,T}(a_1)) = f^{\oplus}_{2,P}(pre_0(a_1)), \ldots,$$

$$f^{\oplus}_{2,P}(pre_0(a_{n-1})) = pre_2(f_{2,T}(a_{n-1})) = pre_2(f_{2,T}(a_n)) = pre_2(t_2).$$

Thus, using commutativity of (2) and the fact that $\_^{\oplus}$ is a functor we obtain

$$\begin{array}{rcl} f'^{\oplus}_{1,P}(pre_1(t_1)) & = & f'^{\oplus}_{1,P}(f^{\oplus}_{1,P}(pre_0(a_0))) \\ & = & (f'_{1,P} \circ f_{1,P})^{\oplus}(pre_0(a_0)) \\ & = & (f'_{2,P} \circ f_{2,P})^{\oplus}(pre_0(a_0)) \\ & = & f'^{\oplus}_{2,P}(f^{\oplus}_{2,P}(pre_0(a_0)) \\ & = & \ldots \\ & = & f'^{\oplus}_{2,P}(f^{\oplus}_{2,P}(pre_0(a_n)) \\ & = & f'^{\oplus}_{2,P}(pre_2(t_2)) \end{array}$$

which means that the two cases lead to the same result. Hence, the definition over functions $f'^{\oplus}_{1,P} \circ pre_1 : T_1 \to P_3^{\oplus}$ and $f'^{\oplus}_{2,P} \circ pre_2 : T_2 \to P_3^{\oplus}$ lead to a well-defined function $pre_3 : T_3 \to P_3^{\oplus}$. The argumentation for the functions $post_3$, $cond_3$ and $type_3$ works analogously.

Well-definedness of AHL-morphisms f'_1 and f'_2. The required compatibilities of the morphisms with the pre, post and firing conditions and type functions of the AHL-nets follow directly from the definition of the respective functions in the AHL-net AN_3, e. g. given $t \in T_1$ we have

$$pre_3(f'_{1,T}(t)) = f'^{\oplus}_{1,T}(pre_1(t))$$

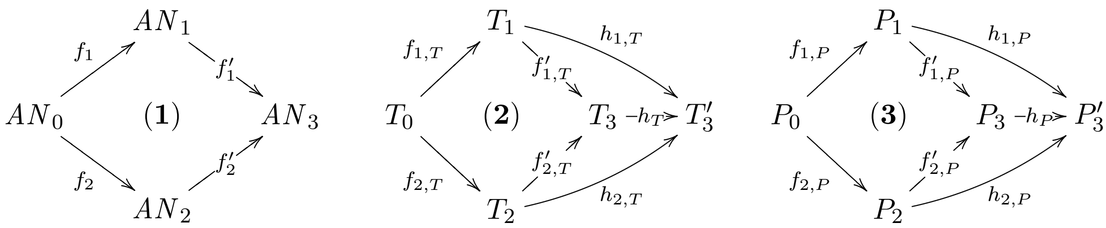
by the definition of pre_3.

□

A.3 Proof of Fact 4.2 (Pushout of AHL-Nets)

The diagram (1) in Def. 4.3 is a *pushout diagram* in the category **AHLNets**, i. e. (1) commutes and it has the following universal property: For all AHL-nets AN'_3 and AHL-morphisms $h_1 : AN_1 \to AN'_3$, $h_2 : AN_2 \to AN'_3$ with $h_1 \circ f_1 = h_2 \circ f_2$ there exists a unique AHL-morphism $h : AN_3 \to AN'_3$ such that $h \circ f'_1 = h_1$ and $h \circ f'_2 = h_2$.

Proof. Given an AHL-net $AN'_3 = (\Sigma, P'_3, T'_3, pre'_3, post'_3, cond'_3, type'_3, A)$ together with morphisms $h_1 : AN_1 \to AN'_3$, $h_2 : AN_2 \to AN'_3$ with $h_1 \circ f_1 = h_2 \circ f_2$. By Def. 4.3 the diagrams (2) and (3) are pushouts in **Sets** leading to a unique morphism $h_P : P_3 \to P'_3$ with $h_P \circ f'_{1,P} = h_{1,P}$ and $h_P \circ f'_{2,P} = h_{2,P}$, and a unique morphism $h_T : T_3 \to T'_3$ with $h_T \circ f'_{1,T} = h_{1,T}$ and $h_T \circ f'_{2,T} = h_{2,T}$.



We define $h := (h_P, h_T)$.

Well-definedness of h. Let $t \in T_3$. We distinguish the following two cases:

- **Case 1.** There is $t_1 \in T_1$ with $f'_{1,T}(t_1) = t$.

$$\begin{aligned}
h_P^{\oplus} \circ pre_3(t) &= h_P^{\oplus} \circ f'^{\oplus}_{1,P} \circ pre_1(t_1)\\
&= (h_P \circ f'_{1,P})^{\oplus} \circ pre_1(t_1)\\
&= h_{1,P}^{\oplus} \circ pre_1(t_1)\\
&= pre'_3 \circ h_{1,T}(t_1)\\
&= pre'_3 \circ h_T \circ f'_{1,T}(t_1)\\
&= pre'_3 \circ h_T(t)
\end{aligned}$$

$$\begin{aligned}
h_P^{\oplus} \circ post_3(t) &= h_P^{\oplus} \circ f'^{\oplus}_{1,P} \circ post_1(t_1)\\
&= (h_P \circ f'_{1,P})^{\oplus} \circ post_1(t_1)\\
&= h_{1,P}^{\oplus} \circ post_1(t_1)\\
&= post'_3 \circ h_{1,T}(t_1)\\
&= post'_3 \circ h_T \circ f'_{1,T}(t_1)\\
&= post'_3 \circ h_T(t)
\end{aligned}$$

$$\begin{aligned}
cond'_3 \circ h_T(t) &= cond'_3 \circ h_T \circ f'_{1,T}(t)\\
&= cond'_3 \circ h_{1,T}(t_1)\\
&= cond_1(t_1)\\
&= cond_3(t)
\end{aligned}$$

- **Case 2.** There is $t_2 \in T_2$ with $f'_{2,T}(t_2) = t$.
  This case works analogously to Case 1.

Now, let $p \in P_3$. Again there are two similar cases and we consider w. l. o. g. the case that there is $p_1 \in P_1$ with $f'_{1,P}(p_1) = p$. Then we have

$$\begin{aligned}
type'_3 \circ h_P(p) &= type'_3 \circ h_P \circ f'_{1,P}(p_1)\\
&= type'_3 \circ h_{1,P}(p_1)\\
&= type_1(p_1)\\
&= type_3(p)
\end{aligned}$$

Hence, $h : AN_3 \to AN'_3$ is a well-defined AHL-morphism and it satisfies $h \circ f'_1 = h_1$ and $h \circ f'_2 = h_2$.

Uniqueness of h. Let $h' = (h'_P, h'_T) : AN_3 \to AN'_3$ be an AHL-morphism with $h' \circ f'_1 = h_1$ and $h' \circ f'_2 = h_2$. Then we have also $h'_T \circ f'_{1,T} = h_{1,T}$ and $h'_T \circ f'_{2,T} = h_{2,T}$ which by the uniqueness of h_T w.r.t. pushout (2) implies that $h'_T = h_T$. Analogously the uniqueness of h_P w.r.t. pushout (3) implies that $h'_P = h_P$ and hence $h' = h$.

□

A.4 Proof of Fact 4.3 (Transformation of AHL-Nets)

Given a production for AHL-nets $p = (L \stackrel{l}{\leftarrow} I \stackrel{r}{\rightarrow} R)$ and a match $m : L \to AN$. The production p is applicable on match m, i. e. there exists a context AHL-net AN_0 in the diagram below, such that (1) is pushout, iff p and m satisfy the gluing condition in **AHLNets**. Then AN_0 is called *pushout complement* of l and m. Moreover, we obtain a unique AN' as pushout object of the pushout (2) in **AHLNets**.

$$\begin{array}{ccccc} L & \xleftarrow{l} & I & \xrightarrow{r} & R \\ {\scriptstyle m}\downarrow & (1) & \downarrow{\scriptstyle c} & & \downarrow{\scriptstyle n} \\ AN & \xleftarrow[d]{} & AN_0 & \longrightarrow & AN' \end{array}$$

If the AHL-net AN_0 exists it is unique up to isomorphism and can be constructed as follows:

- $P_{AN_0} = (P_{AN} \setminus m_P(P_L)) \cup m_P(l_P(P_I))$,
- $T_{AN_0} = (T_{AN} \setminus m_T(T_L)) \cup m_T(l_T(T_I))$,
- $pre_{AN_0} = pre_{AN}|_{T_{AN_0}}$, $post_{AN_0} = post_{AN}|_{T_{AN_0}}$, $cond_{AN_0} = cond_{AN}|_{T_{AN_0}}$ and $type_{AN_0} = type_{AN}|_{P_{AN_0}}$,
- AN_0 has the same data part (Σ, A) as AN;
- $c_P(p) = m_P(l_P(p))$ for $p \in P_I$ and $c_T(t) = m_T(l_T(t))$ for $t \in T_I$, and
- d is an inclusion.

Proof. We show the two directions of the proof seperately.

If. Given the production p and match m such that the gluing condition is satisfied we construct the net AN_0 as described in Fact 4.3.

Well-definedness of AHL-net AN_0. For the well-definedness of AN_0 we have to show that the functions pre_{AN_0}, $post_{AN_0}$, $cond_{AN_0}$ and $type_{AN_0}$ are well-defined.

Well-definedness of $type_{AN_0}$. Given $\Sigma = (S, OP; X)$ for every $p \in P_{AN_0}$ we have to show that $type_{AN_0}(p) \in S$ which holds because for all $p \in P_{AN_0}$ there is

$$type_{AN_0}(p) = type_{AN}|_{P_{AN_0}}(p) = type_{AN}(p) \in S.$$

Well-definedness of pre_{AN_0}. For every $t \in T_{AN_0}$ and $(term, p) \leq pre_{AN_0}(t)$ we have to show that $p \in P_{AN_0}$ and $term \in T_{OP}(X)_{type_{AN_0}(p)}$.
So let $(term, p) \leq pre_{AN_0}(t)$. The case that $p \notin m_P(P_L)$ directly implies that $p \in P_{AN_0}$. So let us consider the case that there is a place $p' \in P_L$ with $m_P(p') = p$. Then we have

$$\begin{aligned} (term, p) \leq pre_{AN_0}(t) &\Leftrightarrow (term, p) \leq pre_{AN}|_{T_{AN_0}}(t) \\ &\Rightarrow (term, p) \leq pre_{AN}(t) \end{aligned}$$

Now for $t \in T_{AN_0}$ there are two possible cases:

- **Case 1.** There is $t \in T_{AN} \setminus m_T(T_L)$.
  Then p' is a dangling point. By satisfaction of the gluing condition p' is also a gluing point, i. e. $p' \in l_P(P_I)$ and therefore $p \in P_{AN_0}$.
- **Case 2.** There is $t \in m_T(l_T(T_I))$.
  This means that there is $t_0 \in T_I$ with $(m \circ l)_T(t_0) = m_T(l_T(t_0)) = t$. Since $(m \circ l)$ is an AHL-morphism which preserves pre conditions there is $p_0 \in P_I$ with $p_0 \in \bullet t_0$ and $m_P(l_P(p_0)) = (m \circ l)_P(p_0) = p$. Thus, there is $p \in P_{AN_0}$.

The fact that $term \in T_{OP}(X)_{type_{AN_0}(p)}$ follows from the fact that $(term, p) \leq pre_{AN}(t)$ and $T_{OP}(X)_{type_{AN_0}(p)} = T_{OP}(X)_{type_{AN}(p)}$ for $p \in P_{AN_0}$.

Well-definedness of $post_{AN_0}$. The proof for $post_{AN_0}$ works analogously to the one for pre_{AN_0}.

Well-definedness of $cond_{AN_0}$. The well-definedness of $cond_{AN_0}$ follows from the fact that AN_0 has the same signature part Σ as the AHL-net AN and $cond_{AN}$ is well-defined.

Well-definedness of morphism $c : I \to AN_0$. The well-definedness of the function c_P follows from the fact that $c_P(P_I) = m_P(l_P(P_I)) \subseteq P_{AN_0}$ and analogously the well-definedness of the function c_T follows from $m_T(l_T(T_I)) \subseteq T_{AN_0}$. The fact that $c = (c_P, c_T)$ is an AHL-morphism can be concluded from the fact that l and m are AHL-morphisms.

Well-definedness of morphism $d : AN_0 \to AN$. The inclusions d_P and d_T are well-defined functions because $P_{AN_0} \subseteq P_{AN}$ and $T_{AN_0} \subseteq T_{AN}$.
It remains to show that d is an AHL-morphism which holds because the pre, post, cond and type functions are restrictions of the corresponding functions in AN to the set of transitions respectively places in AN_0.

Diagram (1) is pushout in the category AHLNets.

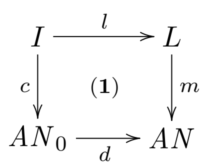


Diagram (1) commutes. For all $p \in P_I$ we have

$$m_P \circ l_P(p) = c_P(p) = d_P \circ c_P(p)$$

and for all $t \in T_I$ we have

$$m_T \circ l_T(t) = c_T(t) = d_T \circ c_T(t)$$

Definition of universal morphism $x : AN \to X$. Let X be an AHL-net and $x_1 : L \to X$, $x_2 : AN_0 \to X$ two AHL-morphisms with $x_1 \circ l = x_2 \circ c$.
We define a morphism $x = (x_P, x_T) : AN \to X$ in the following way:

$$x_P(p) = \begin{cases} x_{1,P}(p') & \text{, if there exists } p' \in P_L : m_P(p') = p; \\ x_{2,P}(p') & \text{, if there exists } p' \in P_{AN_0} : d_P(p') = p. \end{cases}$$

$$x_T(t) = \begin{cases} x_{1,T}(t') & \text{, if there exists } t' \in T_L : m_T(t') = t; \\ x_{2,T}(t') & \text{, if there exists } t' \in T_{AN_0} : d_T(t') = t. \end{cases}$$

Well-definedness of functions x_P and x_T. First, we show that each of the cases in the above definition lead to a unique result. The cases defined via d_P or d_T obviously have a unique result because d is an inclusion. Let us consider the case that for $p \in P_{AN}$ there exist $p_1 \neq p_2 \in P_L$ with $m_P(p_1) = p = m_P(p_2)$.
Then p_1 and p_2 are identification points which by the satifaction of the gluing condition implies that there exist $p_1', p_2' \in P_I$ such that $l_P(p_1') = p_1$ and $l_P(p_2') = p_2$. By definition of morphism c we obtain

$$c_P(p_1') = m_P(l_P(p_1')) = p = m_P(l_P(p_2')) = c_P(p_2')$$

and thus by commutativity $x_1 \circ l = x_2 \circ c$ we have

$$x_{1,P}(p_1) = x_{1,P}(l_P(p_1')) = x_{2,P}(c_P(p_1')) = x_{2,P}(c_P(p_2')) = x_{1,P}(l_P(p_2')) = x_{1,P}(p_2)$$

which means that $x_P(p)$ is uniquely defined.
The case that there exist $t_1 \neq t_2 \in T_L$ with $m_T(t_1) = t = m_T(t_2)$ works analogously.
It remains to show that for every $p \in P_{AN}$, and every $t \in T_{AN}$ respectively, there is one of the cases true.
Let $p \in P_{AN}$ and let us assume that none of the cases is true, i. e. there is no $p_1 \in P_L$ with $m_P(p_1) = p$ and there is no $p_2 \in P_{AN_0}$ with $d_P(p_2) = p$.
Then there is $p \in P_{AN} \setminus m_P(P_L)$ implying that $p \in P_{AN_0}$ with $d_P(p) = p$. This is a contradiction to the assumption that none of the cases is true.
Let $p \in P_{AN}$ and let us assume that both of the cases are true, i. e. there is $p_1 \in P_L$ with $m_P(p_1) = p$ and there is $p_2 \in P_{AN_0}$ with $d_P(p_2) = p$. Since d is an inclusion we have $p_2 = p$.
So the fact that $p \in P_{AN_0}$ by the definition of P_{AN_0} implies that there is $p_0 \in P_I$ with $l_P(p_0) = p_1$. Furthermore there is $c_P(p_0) = p_2$ because (1) commutes. Then from the commutativity $x_1 \circ l = x_2 \circ c$ we obtain

$$\begin{aligned} x_{1,P}(p_1) &= x_{1,P}(l_P(p_0)) \\ &= x_{2,P}(c_P(p_0)) \\ &= x_{2,P}(p_2) \end{aligned}$$

which means that both cases lead to the same result and hence x_P is a well-defined function.
For $t \in T_{AN}$ we obtain analogously that $t \notin m_T(T_L)$ implies that $t \in d_T(T_{AN_0})$ which means that at least one of the cases is true. Analogously to above also the both cases of x_T lead to the same result and hence also x_T is a well-defined function.

Well-definedness of morphism x. We have to show that x is an AHL-morphism.
Let $t \in T_{AN}$.

- **Case 1.** There exists $t' \in T_L$ with $m_T(t') = t$.
  From the fact that m is an AHL-morphism which preserves pre conditions we obtain

  $$\begin{aligned} pre_{AN}(t) &= pre_{AN}(m_T(t)) \\ &= (id_{T_{OP}(X)} \otimes m_P)^{\oplus}(pre_L(t')) \end{aligned}$$

  which means that for all places p in the pre domain of t there is $p' \in P_L$ with $m_P(p') = p$. Therefore we have

  $$\begin{aligned} (id_{T_{OP}(X)} \otimes x_P)^{\oplus} \circ pre_{AN}(t) &= (id_{T_{OP}(X)} \otimes x_P)^{\oplus} \circ pre_{AN} \circ m_T(t') \\ &= (id_{T_{OP}(X)} \otimes x_P)^{\oplus} \circ (id_{T_{OP}(X)} \otimes m_P)^{\oplus} \circ pre_L(t') \\ &= (id_{T_{OP}(X)} \otimes (x_P \circ m_P))^{\oplus} \circ pre_L(t') \\ &= (id_{T_{OP}(X)} \otimes x_{1,P})^{\oplus} \circ pre_L(t') \\ &= pre_X(x_{1,T}(t')) \\ &= pre_X(x_T(t')) \end{aligned}$$

The proof for the post conditions works analogously.
For the firing conditions we obtain

$$\begin{aligned} cond_X \circ x_T(t) &= cond_X \circ x_{1,T}(t') \\ &= cond_L(t') \\ &= cond_{AN} \circ m_T(t') \\ &= cond_{AN}(t) \end{aligned}$$

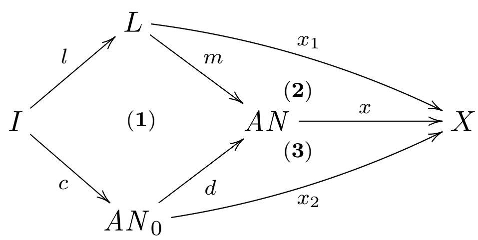
- **Case 2.** There exists $t' \in T_{AN_0}$ with $d_T(t') = t$.
  This case is completely analogous to case 1 due to the symmetric definition of x and the fact that x_2 and d are AHL-morphisms as well as x_1 and m.

Let $p \in P_{AN}$ and let $p' \in P_L$ with $p = m_P(p')$. Then we have

$$\begin{aligned} type_X \circ x_P(p) &= type_X \circ x_{1,P}(p') \\ &= type_L(p') \\ &= type_{AN} \circ m_P(p') \\ &= type_{AN}(p) \end{aligned}$$

The case that there is $p' \in P_{AN_0}$ with $p = d_P(p')$ works analogously.
Hence x is a well-defined AHL-morphism.

Universal property. We have to show the commutativity of (2) and (3) which follows directly from the definition of x.



Uniqueness of x. Let $x' : AN \to X$ be an AHL-morphism with $x' \circ m = x_1$ and $x' \circ d = x_2$. As mentioned above in the proof of the well-definedness of x_P and x_T for every element in AN there is a preimage in L or AN_0.
Let $p \in P_{AN}$.

- **Case 1.** There is p' in P_L with $p = m_P(p')$.

$$x'_P(p) = x'(m_P(p')) = x_{1,P}(p') = x_P(m_P(p')) = x_P(p)$$

- **Case 2.** There is p' in P_{AN_0} with $p = d_P(p')$.

$$x'_P(p) = x'(d_P(p')) = x_{2,P}(p') = x_P(d_P(p')) = x_P(p)$$

So we have that $x'_P = x_P$. The proof for $x'_T = x_T$ works completely analogously due to the similar definition of x_T.

Uniqueness of AN_0. The fact that the pushout complement AN_0 is unique up to isomorphism follows from the uniqueness of pushout complements in $\mathcal{M}$-adhesive categories and the fact that (**AHLNets**, $\mathcal{M}$) with the class $\mathcal{M}$ of all injective AHL-morphisms is an $\mathcal{M}$-adhesive category (see Fact 4.25 and Theorem 4.26(4) in [EEPT06] where the category **AHLNets** is called **AHLNets(SP, A)** and $\mathcal{M}$-adhesive categories are called weak adhesive HLR categories).

Only If. Given pushout (1) in **AHLNets** let us assume that the gluing condition is not satisfied.

- **Case 1.** There is a dangling point which is no gluing point.
  Then there is a transition $t \in T_{AN} \setminus m_T(T_L)$ and a place $p \in P_L \setminus l_P(P_I)$ together with a term $term \in T_{OP}(X)_{type(p)}$ such that

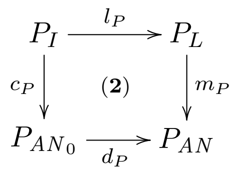
  $$(term, m_P(p)) \leq pre_{AN}(t) \oplus post_{AN}(t).$$

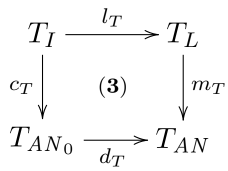
  Due to the uniqueness of pushout complements we can w. l. o. g. assume that AN_0 is constructed as described in Fact 4.3. So the fact that $t \notin m_T(T_L)$ implies $t \in T_{AN_0}$. Moreover the fact that $p \notin l_P(P_I)$ implies that $m_P(p) \notin m_P(l_P(P_I))$ and thus $m_P(p) \notin P_{AN_0}$ by construction of AN_0. So we have

  $$\begin{aligned} & (term, m_P(p)) \leq pre_{AN}(t) \oplus post_{AN}(t) \\ \Rightarrow \quad & (term, m_P(p)) \leq pre_{AN}|_{T_{AN_0}}(t) \oplus post_{AN}|_{T_{AN_0}}(t) \\ \Leftrightarrow \quad & (term, m_P(p)) \leq pre_{AN_0}(t) \oplus post_{AN_0}(t) \end{aligned}$$

  which means that AN_0 has a "dangling arc" because $m_P(p) \notin P_{AN_0}$ and hence AN_0 is not a well-defined AHL-net.

- **Case 2.** There is an identification point which is no gluing point.
  We consider the case that there is $p \neq p' \in P_L$ with $p \notin l_P(P_I)$ and $m_P(p) = m_P(p')$. Due to the uniqueness of pushouts by Fact 4.2 we can w. l. o. g. assume that the pushout (1) is constructed as defined in Def. 4.3 implying pushouts (2) and (3) in **Sets**.

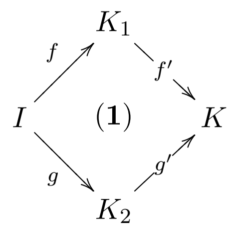
  

  The place p violates the gluing condition for l_P and m_P in **Sets** (see Def. A.4 in [MGE+10]) which by Fact A.7 in [MGE+10] means that also the categorical gluing condition for l_P and m_P is not satisfied. Finally, by Theorem 6.4 in [EEPT06] this contradicts the fact that P_{AN_0} is a pushout complement in (2).
  The proof works analogously for the case that there is $t \neq t' \in T_L$ with $t \notin l_T(T_I)$ and $m_T(t) = m_T(t')$ due to pushout (3) in **Sets**.

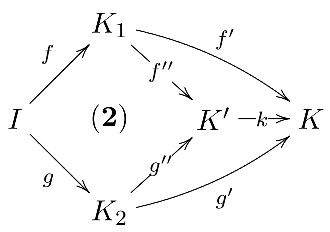
□

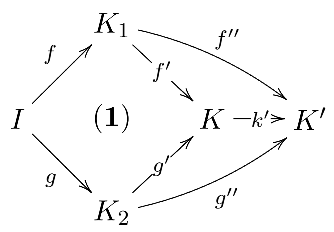
A.5 Proof of Lemma 5.1 (Pushout of AHL-Occurrence Nets)

Given AHL-occurrence nets I, K_1 and K_2 and two AHL-net morphisms $f : I \to K_1$ and $g : I \to K_2$. If (1) is a pushout in **AHLONets** then (1) is also pushout in **AHLNets**.



Proof. Since the category **AHLNets** has pushouts we obtain pushout (2) in **AHLNets**. Then by the fact that **AHLONets** is a subcategory of **AHLNets** by the commutativity of (1) we obtain a unique morphism $k : K' \to K$ with $k \circ f'' = f'$ and $k \circ g'' = g'$.





By Lemma 3.1 we have that K' is an AHL-occurrence net and by the fact that **AHLONets** is full subcategory of **AHLNets** the morphisms f'' and g'' become **AHLONets**-morphisms. So the commutativity of (2) by the universal property of pushout (1) in **AHLONets** implies a unique morphism $k' : K \to K'$ with $k' \circ f' = f''$ and $k' \circ g' = g''$. Now we have

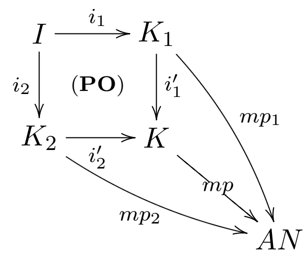
$$k \circ k' \circ f' = k \circ f'' = f' \quad \text{and} \quad k \circ k' \circ g' = k \circ g'' = g'$$

which by the universal property of pushout (1) implies that $k \circ k' = id_K$. Analogously we obtain by the universal property of pushout (2) that $k' \circ k = id_{K'}$ and, thus, k and k' become inverse isomorphisms. Hence, by the uniqueness of pushouts up to isomorphism it follows that (1) is also pushout in **AHLNets**. □

A.6 Proof of Fact 5.2 (Gluing of AHL-Processes)

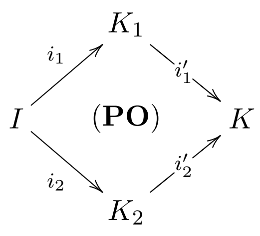
Given AHL-occurrence nets I, K_1, K_2 and two injective AHL-net morphisms $i_1 : I \to K_1$ and $i_2 : I \to K_2$. Then there exists a pushout (PO) in the category **AHLONets** (see Def. 3.3) iff (K_1, K_2) are composable w.r.t. (I, i_1, i_2). The AHL-occurrence net K is then called gluing of K_1 and K_2 along i_1 and i_2, written $K = K_1 +_{(I,i_1,i_2)} K_2$.

In order to extend this gluing construction for AHL-processes in the category **Proc**(**AN**) (see Def. 3.5) one additionally requires AHL-morphisms $mp_1 : K_1 \to AN$ and $mp_2 : K_2 \to AN$ with $mp_1 \circ i_1 = mp_2 \circ i_2$. The pushout (PO) in **AHLONets** then provides a unique morphism $mp : K \to AN$ such that (PO) is also a pushout in **Proc**(**AN**).



Proof. We show the two directions of the proof seperately.

If. Given the AHL-occurrence nets K_1, K_2 and I and morphisms i_1, i_2 as above we construct the pushout (PO) in the category **AHLNets**.



In order to show that (PO) is also a pushout in the full subcategory **AHLONets** it suffices to show that the AHL-net K is an AHL-occurrence net, i. e. K is unary, it has no forward or backward conflicts, and the causal relation $<_K$ is a finitary strict partial order.

Unarity. Let us assume that K is not unary, i. e. there are $p \in P_K$, $t \in T_K$ with

$$(term_1, p) \oplus (term_2, p) \leq pre_K(t) \quad \text{or} \quad (term_1, p) \oplus (term_2, p) \leq post_K(t)$$

Let us consider the case that $(term_1, p) \oplus (term_2, p) \leq pre_K(t)$. Due to the universal property of pushout (PO) there is $a \in \{1, 2\}$ and $t' \in T_{K_a}$ with $i'_{a,T}(t') = t$ and since AHL-morphisms preserve pre and post conditions there is

$$pre_K(t) = (id_{T_{OP}(X)} \otimes i'_{a,P})^{\oplus} \circ pre_{K_a}(t').$$

So the fact that $(term_1, p) \oplus (term_2, p) \leq pre_K(t)$ implies

$$(term_1, p_1) \oplus (term_2, p_2) \leq pre_{K_a}(t')$$

with $i'_1(p_1) = p = i'_2(p_2)$. Since i_1 and i_2 are injective also i'_1 and i'_2 are injective, because (PO) is pushout and thus $p_1 = p_2$ which means that K_a is not unary. This is a contradiction to the assumption that K_a is an AHL-occurrence net.
The case $(term_1, p) \oplus (term_2, p) \leq post_K(t)$ works analogously. Hence K is unary.

No forward conflicts. Let us assume that K has a forward conflict, i. e. there are

$$p \in P_K,\ t_1 \neq t_2 \in T_K \quad \text{with} \quad p \in \bullet t_1 \cap \bullet t_2.$$

- **Case 1:** There is $a \in \{1, 2\}$ such that $t_1, t_2 \in i_{a,T}(T_{K_a})$.
Then we have $t'_1, t'_2 \in T_{K_a}$ with

$$i'_{a,T}(t'_1) = t_1 \quad \text{and} \quad i'_{a,T}(t'_2) = t_2$$

and there is $p' \in P_{K_a}$ with

$$i'_{a,P}(p') = p \quad \text{and} \quad p' \in \bullet t'_1 \cap \bullet t'_2$$

because AHL-morphisms preserve pre conditions. This means that K_a has a forward conflict which contradicts the fact that K_a is assumed to be an AHL-occurrence net.

- **Case 2:** There is $t_1 \in i_{1,T}(T_{K_1})$ and $t_2 \in i_{2,T}(T_{K_2})$.
Then we have $t'_1 \in T_{K_1}, t'_2 \in T_{K_2}$ with

$$i'_{1,T}(t'_1) = t_1 \quad \text{and} \quad i'_{2,T}(t'_2) = t_2$$

and since AHL-morphisms preserve pre conditions there are $p_1 \in P_{K_1}$, $p_2 \in P_{K_2}$ with

$$i'_{1,P}(p_1) = p,\ p_1 \in \bullet t'_1 \quad \text{and} \quad i'_{2,P}(p_2) = p,\ p_2 \in \bullet t'_2.$$

By the fact that K is a pushout object of (PO) this implies a place $p_0 \in P_I$ with

$$i_{1,P}(p_0) = p_1 \quad \text{and} \quad i_{2,P}(p_0) = p_2.$$

- **Case 2.1:** There is $p_0 \in OUT(I)$.
Then the fact that $p_1 \in \bullet t'_1$ means that $i_{1,P}(p_0) \notin OUT(K_1)$ which by the composability of (K_1, K_2) w.r.t. (I, i_1, i_2) implies that $i_{2,P}(p_0) \in OUT(K_2)$ contradicting the fact that $i_{2,P}(p_0) = p_2 \in \bullet t'_2$.

- **Case 2.2:** There is $p_0 \notin OUT(I)$.
  This means that there is $t_0 \in T_I$ with $p_0 \in \bullet t_0$. By the fact that i_1 is an AHL-morphism which preserves pre conditions we have $p_1 \in \bullet i_{1,T}(t_0)$ which together with the fact that $p_1 \in \bullet t'_1$ means that $i_{1,T}(t_0) = t'_1$ because K_1 has no forward conflicts. Analogously due to the fact that also K_2 has no forward conflict we obtain that $i_{2,T}(t_0) = t'_2$. Thus, by commutativity of (PO) we have

  $$t_1 = i'_{1,T}(t'_1) = i'_{1,T}(i_{1,T}(t_0)) = i'_{2,T}(i_{2,T}(t_0)) = i'_{2,T}(t'_2) = t_2$$

  which contradicts the assumption that $t_1 \neq t_2$.

Hence K has no forward conflict.

No backward conflicts. Let us now assume that K has a backward conflict, i. e. there is

$$p \in P_K,\ t_1 \neq t_2 \in T_K \quad \text{with} \quad p \in t_1 \bullet \cap t_2 \bullet$$

- **Case 1:** There is $a \in \{1,2\}$ such that $t_1, t_2 \in i_{a,T}(T_{K_a})$.
  Then we have $t'_1, t'_2 \in T_{K_a}$ with

  $$i'_{a,T}(t'_1) = t_1 \quad \text{and} \quad i'_{a,T}(t'_2) = t_2$$

  and there is $p' \in P_{K_a}$ with

  $$i'_{a,P}(p') = p \quad \text{and} \quad p' \in t'_1 \bullet \cap t'_2 \bullet$$

  because AHL-morphisms preserve post conditions.
  This means that K_a has a backward conflict contradicting the fact that K_a is assumed to be an AHL-occurrence net.
- **Case 2:** There is $t_1 \in i_{1,T}(T_{K_1})$ and $t_2 \in i_{2,T}(T_{K_2})$.
  Then we have $t'_1 \in T_{K_1}, t'_2 \in T_{K_2}$ with

  $$i'_{1,T}(t'_1) = t_1 \quad \text{and} \quad i'_{2,T}(t'_2) = t_2$$

  and since AHL-morphisms preserve post conditions there are $p_1 \in P_{K_1}$, $p_2 \in P_{K_2}$ with

  $$i'_{1,P}(p_1) = p,\ p_1 \in t'_1 \bullet \quad \text{and} \quad i'_{2,P}(p_2) = p,\ p_2 \in t'_2 \bullet .$$

  Due to the fact that K is pushout object of (PO) this implies $p_0 \in P_I$ with

  $$i_{1,P}(p_0) = p_1 \quad \text{and} \quad i_{2,P}(p_0) = p_2.$$

- **Case 2.1:** There is $p_0 \in IN(I)$.
  Then the fact that $p_1 \in t'_1\bullet$ means that $i_{1,P}(p_0) \notin IN(K_1)$ which by the composability of (K_1, K_2) w.r.t. (I, i_1, i_2) implies that $i_{2,P}(p_0) \in IN(K_2)$ contradicting the fact that $i_{2,P}(p_0) = p_2 \in t'_2\bullet$.
- **Case 2.2:** There is $p_0 \notin IN(I)$.
  Then there is $t_0 \in T_I$ with $p_0 \in t_0\bullet$ and by the fact that i_1 is an AHL-morphism which preserves post conditions we have $p_1 = i_{1,P}(p_0) \in i_{1,T}(t_0)\bullet$. Since K_1 has no backward conflicts and there is $p_1 \in t'_1\bullet$ it follows that $i_{1,T}(t_0) = t'_1$. Analogously we obtain that $i_{2,T}(t_0) = t'_2$ because also K_2 has no backward conflict. Now, by the commutativity of (PO) we have

  $$t_1 = i'_{1,T}(t'_1) = i'_{1,T}(i_{1,T}(t_0)) = i'_{2,T}(i_{2,T}(t_0)) = i'_{2,T}(t'_2) = t_2$$

  which contradicts the assumption that $t_1 \neq t_2$.

Hence K has no backward conflict.

Finitary strict partial order. We have to show that $<_K$ is finitary and irreflexive. Due to the fact that AHL-morphisms preserve pre and post conditions we obtain the causal relation of $<_K$ as the transitive closure of

$$\bigcup_{a\in\{1,2\}} \{(i'_a(x), i'_a(y)) \mid x, y \in P_{K_a} \uplus T_{K_a}, x <_{K_a} y\}$$

This means that elements which are causal related in K_1 or K_2 are also causal related in K. Additionally it is possible that elements in the net K are related due to the gluing of one or more elements.
Moreover, if for two interface elements $x_0, y_0 \in P_I \uplus T_I$ the images of these elements are causal related in K, i.e. $i'_1(i_1(x_0)) <_K i'_1(i_1(y_0))$, then there is $x_0 <_{(i_1,i_2)} y_0$. We prove that fact because we need it in the following.
Let $x_0, y_0 \in P_I \uplus T_I$ with $i'_1(i_1(x_0)) <_K i'_1(i_1(y_0))$. Then there is $a \in \{1,2\}$ such that either there is $i_a(x_0) <_{K_a} i_a(y_0)$ or there is $z_0 \in P_I \uplus T_I$ with $i_a(x_0) <_{K_a} i_a(z_0)$ and $i'_1(i_1(x_0)) <_K i'_1(i_1(z_0)) <_K i'_1(i_1(y_0))$. This recursively leads to the fact that $x_0 <_{(i_1,i_2)} y_0$ because the induced causal relation is transitive and finitary.

Irreflexivity. Let us assume that $<_K$ is not irreflexive, i.e. there exists $x \in P_K \uplus T_K$ s.t. $x <_K x$. This means that there is a cycle in K and hence because of the bipartite structure of AHL-nets there exists $x' \in P_K \uplus T_K$ with $x <_K x'$ and $x' <_K x$.

Let us assume that there is no $z \in P_I \uplus T_I$ with $x <_K i'_1(i_1(z)) <_K x'$, i.e. the causal relation of x and x' is not the result of a gluing but is directly obtained from a causal relation in K_1 or K_2. Then there is $a \in \{1,2\}$ and $y, y' \in P_{K_a} \uplus T_{K_a}$ s.t. $i'_a(y) = x$ and $i'_a(y') = x'$ and we have $y <_{K_a} y'$ and $y' <_{K_a} y$. Then the transitivity of $<_{K_a}$ implies $y <_{K_a} y$ which contradicts the fact that $<_{K_a}$ is irreflexive because K_a is an AHL-occurrence net.

So there is $z \in P_I \uplus T_I$ with $x <_K i'_1(i_1(z)) <_K x'$. Due to the transitivity of $<_K$ there is $i'_1(i_1(z)) <_K i'_1(i_1(z))$ because

$$i'_1(i_1(z)) <_K x' <_K x <_K i'_1(i_1(z)).$$

As shown above this implies $z <_{(i_1,i_2)} z$, contradicting the fact that by the composability of K_1 and K_2 w.r.t. (I, i_1, i_2) the induced causal relation $<_{(i_1,i_2)}$ is irreflexive. Hence $<_K$ is irreflexive.

Finitariness. We define $P(x)$ as the set of all predecessors of x in K, i.e.

$$P(x) = \{x' \in P_K \uplus T_K \mid x' <_K x\}$$

and $PI(x)$ as the set of all interface elements which have an image that is predecessor of x, i.e.

$$PI(x) = \{z \in P_I \uplus T_I \mid i'_1(i_1(z)) <_K x\}.$$

In order to show the finitariness of $<_K$ we have to show that for every $x \in P_K \uplus T_K$ there is a finite number $n \in \mathbb{N}$ such that $|P(x)| = n$.

Let us first assume that there is $x \in P_K \uplus T_K$ which has an infinite set of predecessors which are the images of interface elements, i.e. $PI(x)$ is an infinite set. Then we also have that $P(x)$ is an infinite set because $i'_1(i_1(PI(x))) \subseteq P(x)$ and i_i, i'_1 are injective. Due to the finitariness of $<_{K_1}$ and $<_{K_2}$ there are finite many elements which have a causal relation to x directly obtained from the net K_1 resp. K_2.
So there is $y \in PI(x)$ such that for $\tilde{x} = i'_1(i_1(y))$ there is $PI(\tilde{x})$ an infinite set.
For every $z \in PI(\tilde{x})$ there is $z <_{(i_1,i_2)} y$ implying that the induced causal relation $<_{(i_1,i_2)}$ is not

finitary. This contradicts the fact that K_1 and K_2 are composable w.r.t. (I, i_1, i_2).
Thus, for every $x \in P_K \uplus T_K$ there is $m \in \mathbb{N}$ such that $|PI(x)| \leq m$ which allows us to do a mathematical induction to show that for all upper bounds $m \in \mathbb{N}$ the following property holds:

For every $x \in P_K \uplus T_K$ with $PI(x) \leq m$ there exists $n \in \mathbb{N}$ such that $|P(x)| = n$.

Induction Basis. $m = 0$

Let $x \in P_K \uplus T_K$ with $|PI(x)| \leq m = 0$.
Due to the fact that (PO) is a pushout in **AHLNets** i'_1 and i'_2 are jointly surjective and hence there is $a \in \{1, 2\}$ and $x' \in P_{K_a} \uplus T_{K_a}$ with $i'_a(x') = x$. Then the fact that there is no element $y \in P_I \uplus T_I$ with $i'_1(i_1(y)) <_K x$ implies that the causal relation of all predecessors of x is directly derived from the net K_a, i.e. for every $z \in P_K \uplus T_K$ with $z <_K x$ there is $z' \in P_{K_a} \uplus T_{K_a}$ with

$$i'_a(z') = z \quad \text{and} \quad z' <_{K_a} x'$$

because AHL-morphisms preserve pre and post conditions.

Due to the finitariness of $<_{K_a}$ there is a finite number of predecessors $z' <_{K_a} x'$ and hence there is a finite number $n \in \mathbb{N}$ of elements $z \in P_K \uplus T_K$ with $z = i'_a(z') <_K i'_a(x') = x$, i.e. $|P(x)| = n$.

Induction Hypothesis.

For $m \in \mathbb{N}$ it holds that for every $x \in P_K \uplus T_K$ with $|PI(x)| \leq m$ there exists $n \in \mathbb{N}$ such that $|P(x)| = n$.

Induction Step.

Let $x \in P_K \uplus T_K$ with $|PI(x)| \leq m + 1$. Let us assume that x has an infinite number of predecessors, i.e. there is no $n \in \mathbb{N}$ such that $|P(x)| = n$.

As mentioned in the induction basis there is a finite number of elements y with $y <_K x$ for which there is no $z \in PI(x)$ such that $y <_K i'_1(i_1(z))$. Hence there is $z \in PI(x)$ such that $\tilde{x} = i'_1(i_1(z))$ has an infinite number of predecessors.

Due to the irreflexivity of $<_K$ there is $PI(\tilde{x}) \subseteq PI(x) \setminus \{z\}$ and thus $|PI(\tilde{x})| \leq m$ which by the induction hypothesis implies that there exists $\tilde{n} \in \mathbb{N}$ such that $|P(\tilde{x})| = \tilde{n}$ contradicting the fact that $\tilde{x}$ has an infinite number of predecessors.

So we have for $x \in P_K \uplus T_K$ with $m \in \mathbb{N}$ such that $|PI(x)| \leq m$ that there is a finite number $n \in \mathbb{N}$ of predecessors of x in K.
Let $m \in \mathbb{N}$ such that

$$m = \max_{x \in P_K \uplus T_K} |PI(x)|$$

Then we have that for every $x \in P_K \uplus T_K$ there is $|PI(x)| \leq m$ which implies that there is $n_x \in \mathbb{N}$ such that $|P(x)| = n_x$. Hence $<_K$ is also finitary.

Only If. Given the pushout diagram (PO) in the category **AHLONets**. By Lemma 5.1 (PO) is also a pushout in the category **AHLNets**. We have to show that (K_1, K_2) are composable w.r.t. (I, i_1, i_2).

First condition. We have to show that $\forall\, x \in IN(I) : i_1(x) \notin IN(K_1) \Rightarrow i_2(x) \in IN(K_2)$. Let $x \in IN(I)$ with $i_1(x) \notin IN(K_1)$ and let us assume that there is $i_2(x) \notin IN(K_2)$.

Then $i_1(x)$ and $i_2(x)$ both are in the post domain of transitions, i.e. there are $t_1 \in T_{K_1}$ and $t_2 \in T_{K_2}$ such that $i_1(x) \in t_1\bullet$ and $i_2(x) \in t_2\bullet$. Since AHL-morphisms preserve post conditions there is

$$i'_1(i_1(x)) \in i'_1(t_1)\bullet \quad \text{and} \quad i'_2(i_2(x)) \in i'_1(t_2)\bullet$$

and due to the fact that (PO) commutes there is $i'_1(i_1(x)) = i'_2(i_2(x))$ which implies

$$i'_1(i_1(x)) \in i'_1(t_1) \bullet \cap i'_2(t_2) \bullet .$$

Since K is an AHL-occurrence net it has no backward conflict implying that $i'_1(t_1) = i'_2(t_2)$. So due to the pushout property there is $t_0 \in T_I$ with

$$i_1(t_0) = t_1 \quad \text{and} \quad i_2(t_0) = t_2$$

Then by the fact that $i_1(x) \in i_1(t_0)\bullet$ together with the fact that i_1 is an AHL-morphism which preserves post domains it follows that $x \in t_0\bullet$. This contradicts the fact that $x \in IN(I)$. Hence, there is $i_2(x) \in IN(K_2)$.

Second condition. We have to show that $\forall\, x \in OUT(I) : i_1(x) \notin OUT(K_1) \Rightarrow i_2(x) \in OUT(K_2)$. Let $x \in OUT(I)$ with $i_1(x) \notin OUT(K_1)$ and let us assume that there is $i_2(x) \notin OUT(K_2)$.

Then $i_1(x)$ and $i_2(x)$ both are in the pre domain of transitions, i. e. there are $t_1 \in T_{K_1}$ and $t_2 \in T_{K_2}$ such that $i_1(x) \in \bullet t_1$ and $i_2(x) \in \bullet t_2$. Since AHL-morphisms preserve pre conditions there is

$$i'_1(i_1(x)) \in \bullet i'_1(t_1) \quad \text{and} \quad i'_2(i_2(x)) \in \bullet i'_1(t_2)$$

and by commutativity of (PO) we have $i'_1(i_1(x)) = i'_2(i_2(x))$ which implies

$$i'_1(i_1(x)) \in \bullet i'_1(t_1) \cap \bullet i'_2(t_2)$$

Since K is an AHL-occurrence net it has no forward conflict implying that $i'_1(t_1) = i'_2(t_2)$. So due to the pushout property there is $t_0 \in T_I$ with

$$i_1(t_0) = t_1 \text{ and } i_2(t_0) = t_2$$

Then by the fact that $i_1(x) \in \bullet i_1(t_0)$ together with the fact that i_1 is an AHL-morphism which preserves pre domains it follows that $x \in \bullet t_0$. This contradicts the fact that $x \in OUT(I)$. Hence, there is $i_2(x) \in OUT(K_2)$.

Induced causal relation is a finitary strict partial order. Let $x, y \in P_I \uplus T_I$ with $x \prec_{(i_1,i_2)} y$. Then by the definition of $\prec_{(i_1,i_2)}$ there is

$$i_1(x) <_{K_1} i_1(y) \quad \text{or} \quad i_2(x) <_{K_2} i_2(y)$$

and by the fact that $i'_1 \circ i_1 = i'_2 \circ i_2$ we have

$$i'_1 \circ i_1(x) <_K i'_1 \circ i_1(y)$$

because AHL-morphisms preserve pre and post conditions.
Since $<_K$ is transitive we have also for the transitive closure $<_{(i_1,i_2)}$ of $\prec_{(i_1,i_2)}$ that $x <_{(i_1,i_2)} y$ implies $i'_1 \circ i_1(x) <_K i'_1 \circ i_1(y)$.

Let us assume that $<_{(i_1,i_2)}$ is not finitary, i. e. there is $y \in P_I \uplus T_I$ with an infinite set of predecessors

$$S = \{x \in P_I \uplus T_I \mid x <_{(i_1,i_2)} y\}$$

leading to an infinite set

$$S' = \{x \in P_I \uplus T_I \mid i'_1 \circ i_1(x) <_K i'_1 \circ i_1(y)\}$$

which contradicts the fact that $<_K$ is finitary.

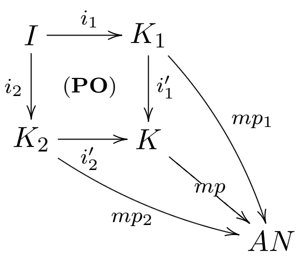
Let us assume that $<_K$ is not irreflexive, i. e. there is $x \in P_I \uplus T_I$ with $x <_{(i_1,i_2)} x$. Then there is

$$i'_1 \circ i_1(x) <_K i'_1 \circ i_1(x)$$

contradicting the fact that $<_K$ is irreflexive.

So we have that the induced causal relation $<_{(i_1,i_2)}$ is finitary and irreflexive and hence it is a finitary strict partial order.

Extension to Processes.
Given the pushout (PO) and additional AHL-morphisms $mp_1 : K_1 \to AN$ and $mp_2 : K_2 \to AN$ with $mp_1 \circ i_1 = mp_2 \circ i_2$.

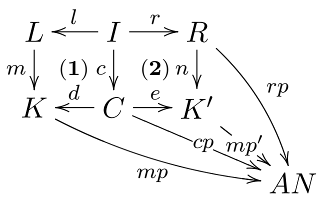


Then we also have a morphism $mp_0 : I \to AN$ defined by $mp_0 := mp_1 \circ i_1 = mp_2 \circ i_2$. Moreover the pushout property of (PO) implies a unique morphism $mp : K \to AN$ such that (PO) is also a pushout in the slice category **AHLNets** $\setminus AN$. As shown above the composability of K_1 and K_2 w.r.t. (I, i_1, i_2) implies that K is an AHL-occurrence net. Hence, $mp : K \to AN$ is an AHL-process which implies that (PO) is also pushout in the full subcategory **Proc**(**AN**) $\subseteq$ **AHLNets** $\setminus AN$ of AHL-processes. □

A.7 Proof of Theorem 5.3 (Direct Transformation of AHL-Processes)

Given a production for AHL-processes $p : L \overset{l}{\leftarrow} I \overset{r}{\rightarrow} R$ and an AHL-occurrence net K together with an injective morphism $m : L \to K$. Then the direct transformation of AHL-occurrence nets with pushouts (1) and (2) in **AHLONets** exists iff p satisfies the transformation condition for AHL-processes under m.

In order to extend this construction for AHL-processes in the category **Proc**(**AN**) one additionally requires AHL-morphisms $mp : K \to AN$ and $rp : R \to AN$ with $mp \circ m \circ l = rp \circ r$. Then the pushout (1) in **AHLONets** is a pushout of $mp \circ m$ and $cp = mp \circ d$ in **Proc**(**AN**), and the pushout (2) in **AHLONets** provides a unique morphism $mp' : K' \to AN$ such that mp' is pushout of cp and rp in **Proc**(**AN**) according to Fact 5.2.



Proof. First, we prove the following lemma which states the equivalence of the gluing relation for a given production and match and the induced causal relation of the right-hand side of the production and the context net in **AHLNets**.

Lemma A.1 (Gluing Relation Lemma)**.** *Given a production for AHL-occurrence nets* $p : L \xleftarrow{l} I \xrightarrow{r} R$*, a match* $m : L \to K$ *where* K *is an AHL-occurrence net, and pushout (1) in* **AHLNets**.

Then the gluing relation $<_{(p,m)}$ *is exactly the induced causal relation of* C *and* R *w.r.t.* (I, c, r)*, i. e.* $<_{(p,m)} = <_{(c,r)}$.

$$\begin{array}{ccccc} L & \xleftarrow{l} & I & \xrightarrow{r} & R \\ {\scriptstyle m}\downarrow & (1) & \downarrow{\scriptstyle c} & & \\ K & \xleftarrow[d]{} & C & & \end{array}$$

Proof. We define a relation $\prec_C \subseteq (P_C \times T_C) \uplus (T_C \times P_C)$ as follows:

$$\prec_C = \{(p,t) \in P_C \times T_C \mid p \in \bullet t\} \cup \{(t,p) \in T_C \times P_C \mid p \in t\bullet\}$$

The relation $\prec_C$ describes the direct causal relationship of the elements in C, i. e. the causal relation $<_C$ is the transitive closure of $\prec_C$. We show that $\prec_{(K,m)} = \prec_C$ by showing that there is a subset relation in both directions.

Direction 1 ($\prec_{(K,m)} \subseteq \prec_C$). Let $x, y \in P_K \uplus (T_K \setminus m_T(T_L))$ with $x \prec_{(K,m)} y$. Due to the structure of petri nets there are two possible cases:

- **Case 1.** There is $x \in P_K$ and $y \in T_K \setminus m_T(T_L)$.
  Due to the construction of C there is $y \in T_C$. Furthermore there is $term \in T_{OP}(X)_{type_K(x)}$ such that

$$\begin{aligned} (term, x) \le pre_K(y) &\Leftrightarrow (term, x) \le pre_K|_{T_C}(y) \\ &\Leftrightarrow (term, x) \le pre_C(y) \end{aligned}$$

  and hence $x \prec_C y$.
- **Case 2.** There is $x \in T_K \setminus m_T(T_L)$ and $y \in P_K$.
  In this case we have $x \in T_C$ and there is $term \in T_{OP}(X)_{type_K(x)}$ such that

$$\begin{aligned} (term, y) \le post_K(x) &\Leftrightarrow (term, y) \le post_K|_{T_C}(x) \\ &\Leftrightarrow (term, y) \le post_C(x) \end{aligned}$$

  and hence $x \prec_C y$.

Direction 2 ($\prec_C \subseteq \prec_{(K,m)}$). Let $x, y \in P_C \uplus T_C$ with $x \prec_C y$. Again we distinguish the two possible cases:

Case 1. There is $x \in P_C$ and $y \in T_C$.
Then there is $term \in T_{OP}(X)_{type_C(x)}$ such that $(term, x) \le pre_C(y)$. Since AHL-morphisms preserve pre conditions and d is an inclusion we have

$$\begin{aligned} (term, x) \le pre_C(y) &\Leftrightarrow (term, x) \le d^{\oplus} \circ pre_C(y) \\ &\Leftrightarrow (term, x) \le pre_K(d(y)) \\ &\Leftrightarrow (term, x) \le pre_K(y) \end{aligned}$$

So the fact that $T_C = T_K \setminus m_T(T_L)$ implies $x \prec_{(K,m)} y$.

Case 2. There is $x \in T_C$ and $y \in P_C$.
Then there is $term \in T_{OP}(X)_{type_C(x)}$ such that $(term, x) \le post_C(y)$. Since AHL-morphisms preserve not only pre but also post conditions we obtain analogously to Case 1 that $x \prec_{(K,m)} y$.

So we have that $\prec_{(K,m)} = \prec_C$ and since $<_{(K,m)}$ is the transitive closure of $\prec_{(K,m)}$ and $<_C$ is the transitive closure of $\prec_C$ it follows that $<_{(K,m)} = <_C$.
Furthermore we can use the inclusion d to obtain from the commutativity of (1) that

$$m \circ l(x) = d \circ c(x) = c(x).$$

So let $\prec_{(c,r)} \subseteq (P_I \times T_I) \uplus (T_I \times P_I)$ be the relation defined by

$$\prec_{(c,r)} \; = \; \{(x,y) \mid c(x) <_C c(y) \vee r(x) <_R r(y)\}$$

then we have

$$\begin{aligned}\prec_{(p,m)} &= \{(x,y) \in (P_I \times T_I) \uplus (T_I \times P_I) \mid m \circ l(x) <_{(K,m)} m \circ l(y) \vee r(x) <_R r(y)\} \\ &= \{(x,y) \in (P_I \times T_I) \uplus (T_I \times P_I) \mid c(x) <_C c(y) \vee r(x) <_R r(y)\} \\ &= \prec_{(c,r)}\end{aligned}$$

and since $<_{(p,m)}$ is the transitive closure of $\prec_{(p,m)}$ and $<_{(c,r)}$ is the transitive closure of $\prec_{(c,r)}$ we have $<_{(p,m)} = <_{(c,r)}$.

□

Now we show that the pushouts (1) and (2) below exist in **AHLONets** if and only if the production p under m satisfies the transformation condition for AHL-occurrence nets.

$$\begin{array}{ccccc} L & \xleftarrow{l} & I & \xrightarrow{r} & R \\ {\scriptstyle m}\downarrow & (1) & \downarrow{\scriptstyle c} & (2) & \downarrow{\scriptstyle n} \\ K & \xleftarrow{d} & C & \xrightarrow{e} & K' \end{array}$$

If. Given production $p : L \xleftarrow{l} I \xrightarrow{r} R$ satisfying the transformation condition for AHL-processes under match m. Since this implies that p satisfies the the gluing condition for AHL-nets by Theorem 4.3 there exist pushouts (1) and (2) in **AHLNets**. We have to show that (1) and (2) are also pushouts in the category **AHLONets** of AHL-occurrence nets.

Pushout (1). From AHL-occurrence net K and AHL-morphism $d : C \to K$ it follows by Lemma 3.1 that also C is an AHL-occurrence net. So we have that all objects and morphisms in pushout (1) are in the full subcategory **AHLONets** $\subseteq$ **AHLNets** which means that (1) is also a pushout in **AHLONets**.

Pushout (2). We have to show that (C, R) are composable w.r.t. (I, c, r), i. e.

1. c is injective,
2. $\forall\, x \in IN(I) : c(x) \notin IN(C) \Rightarrow r(x) \in IN(R)$ and
   $\forall\, x \in OUT(I) : c(x) \notin OUT(C) \Rightarrow r(x) \in OUT(R)$, and
3. the induced causal relation $<_{(c,r)}$ is a finitary strict partial order

Part 1. Due to the fact that (**AHLNets**, $\mathcal{M}$) with the class $\mathcal{M}$ of all injective AHL-morphisms is an $\mathcal{M}$-adhesive category (see [EEPT06]) pushout (1) along $\mathcal{M}$-morphism l is also a pullback which preserves monomorphisms. Thus, monomorphism m implies that c is a monomorphism and since the monomorphisms in **AHLNets** are exactly the injective morphisms it follows that c is injective.

Part 2. From pushout (1) in **AHLONets** of injective morphisms l and c it follows by Fact 5.2 that (L, R) are composable w.r.t. (I, l, c).

Let $x \in IN(I)$ and $c(x) \notin IN(C)$ then by the composability of (L, C) w.r.t. (I, l, c) follows that $l(x) \in IN(L)$. The fact that $c(x) \notin IN(C)$ implies $t \in T_C$ with $c(x) \in t\bullet$ and $g_2 \circ c(x) \in g_2(t)\bullet$ because AHL-morphisms preserve post conditions. Due to the commutativity of (1) there is $m \circ l(x) = g_2 \circ c(x)$ which means that $m \circ l(x) \notin IN(K)$ because $m \circ l(x) \in g_2(t)\bullet$.

So there is $x \in InP$ and the fact that production p satisfies the transformation condition for AHL-processes implies that $r(x) \in IN(R)$.

Now, let $x \in OUT(I)$ and $c(x) \notin OUT(C)$ then by the composability of (L, C) w.r.t. (I, l, c) follows that $l(x) \in OUT(L)$. The fact that $c(x) \notin OUT(C)$ implies $t \in T_C$ with $c(x) \in \bullet t$ and $g_2 \circ c(x) \in \bullet g_2(t)$ because AHL-morphisms preserve pre conditions. Due to the commutativity of (1) there is $m \circ l(x) = g_2 \circ c(x)$ which means that $m \circ l(x) \notin OUT(K)$ because $m \circ l(x) \in \bullet g_2(t)$.

So there is $x \in OutP$ and the fact that production p satisfies the transformation condition for AHL-processes implies that $r(x) \in OUT(R)$.

Part 3. The fact that the gluing relation $<_{(p,m)}$ of p und m is a finitary strict partial order implies that the induced causal relation $<_{(c,r)}$ is a finitary strict partial order because by Lemma A.1 there is $x <_{(p,m)} y \Leftrightarrow x <_{(c,r)} y$.

Thus (C, R) are composable w.r.t. (I, c, r) leading to the existence of pushout (2) in **AHLONets**.

Only If. Given pushouts (1) and (2) in **AHLONets**. We have to show that the transformation condition for AHL-occurrence nets (see Def. 5.5) is satisfied by production p under match m.

Gluing condition. By Lemma 5.1 pushouts (1) and (2) in **AHLONets** are also pushouts in **AHLNets** which by Fact 4.3 implies that the gluing condition is satisfied.

Gluing relation is finitary strict partial order. By Fact 5.2 pushout (2) in **AHLONets** implies that (C, R) are composable w.r.t. (I, c, r) which means that $<_{(c,r)}$ is a finitary strict partial order. Due to Lemma A.1 we know that $<_{(c,r)} = <_{(p,m)}$ which means that also $<_{(p,m)}$ is a finitary strict partial order.

In and out places. Due to the uniqueness of pushout complements in $\mathcal{M}$-adhesive categories we can w. l. o. g. assume that the context net C is constructed as given in Fact 4.3.

Let $x \in InP$ which means that $x \in IN(I)$ with $l_P(x) \in IN(L)$ and $m_P \circ l_P(x) \notin IN(K)$. The fact that $m_P \circ l_P(x) \notin IN(K)$ implies that there is $t \in T_K$ with $m_P \circ l_P(x) \in t\bullet$.

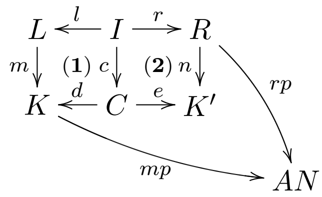
Let us assume that there is $t' \in T_L$ with $m_T(t') = t$. Then from the fact that m is an AHL-morphism follows that $l_P(x) \in t'\bullet$ because AHL-morphisms preserve post conditions. This contradicts the fact that $l_P(x) \in IN(K)$ and thus $t \notin m_T(T_L)$ which means that $t \in T_K \setminus m_T(T_L)$. Then by the construction of T_C it follows that $t \in T_C$.

Moreover, we have $c_P(x) = m_P \circ l_P(x) \in t\bullet$ which means that $c_P(x) \notin IN(C)$ This implies that $r(x) \in IN(R)$ due to the composability of (C, R) w.r.t. (I, c, r) given by pushout (2) in **AHLONets** by Fact 5.2.

Now, let $x \in OutP$ which means that $x \in OUT(I)$ with $l_P(x) \in OUT(L)$ and $m_P \circ l_P(x) \notin OUT(K)$. Then $m_P \circ l_P(x) \notin OUT(K)$ implies that there is $t \in T_K$ with $m_P \circ l_P(x) \in \bullet t$. Again, the assumption that $t' \in T_L$ with $m_T(t') = t$ leads to a contradiction which means that $t \in T_K \setminus m_T(T_L)$. Then by the construction of T_C follows that $t \in T_C$ and we have $c_P(x) = m_P \circ l_P(x) \in \bullet t$ which means that $c_P(x) \notin OUT(C)$ and hence $r(x) \in OUT(R)$ by composability of (C, R) w.r.t. (I, c, r).

Extension to Processes.

Given pushouts (1) and (2) in **AHLONets** and additional morphisms $mp : K \to AN$ and $rp : R \to AN$ with $mp \circ m \circ l = rp \circ r$.



Since L, C and I are AHL-occurrence nets we obtain AHL-processes by composition of AHL-morphisms $lp := mp \circ m : L \to AN$, $cp := mp \circ d : C \to AN$ and $ip := mp \circ m \circ l = mp \circ d \circ c : I \to AN$ such that (1) is a commuting diagram in **Proc(AN)**.

By Lemma 5.1 pushout (1) in **AHLONets** is also a pushout in **AHLNets** and thus by construction of pushouts in slice categories it is also a pushout in **AHLNets** $\setminus AN$. Hence, due to the fact that *lp*, *cp*, *ip* and *mp* are AHL-processes we have that (1) is a pushout in the full subcategory **Proc(AN)** $\subseteq$ **AHLNets** $\setminus AN$.

Finally, we have

$$cp \circ c = mp \circ d \circ c = mp \circ m \circ l = rp \circ r$$

which by Fact 5.2 implies a unique morphism $mp' : K' \to AN$ such that (2) is also a pushout in **Proc(AN)**. □

A.8 Proof of Theorem 7.1 (Amalgamation Theorem for AHL-Processes)

Given the gluing (pushout) $AN_3 = AN_1 +_{AN_0} AN_2$ of AHL-nets in (PO) of Fig. 12 with injective f_1, f_2 then we have:

1. **Composition Construction.** Let $mp_i : K_i \to AN_i$ be AHL-processes for $i \in \{0, 1, 2\}$ such that mp_1 and mp_2 agree on mp_0, then the amalgamation $mp_3 = mp_1 +_{mp_0} mp_2$ exists and is an AHL-process $mp_3 : K_3 \to AN_3$.
2. **Decomposition Construction.** Let $mp_3 : K_3 \to AN_3$ be an AHL-process and let mp_1, mp_2 be restrictions of mp_3 along g_1 resp. g_2, and mp_0 restriction of mp_1 along f_1. Then mp_3 can be represented as amalgamation $mp_3 = mp_1 +_{mp_0} mp_2$.
3. **Bijective Correspondence.** There are composition and decomposition functions

   $$Comp : Proc(AN_1 \overset{f_1}{\leftarrow} AN_0 \overset{f_2}{\rightarrow} AN_2) \to Proc(AN)$$

   and

   $$Decomp : Proc(AN) \to Proc(AN_1 \overset{f_1}{\leftarrow} AN_0 \overset{f_2}{\rightarrow} AN_2)$$

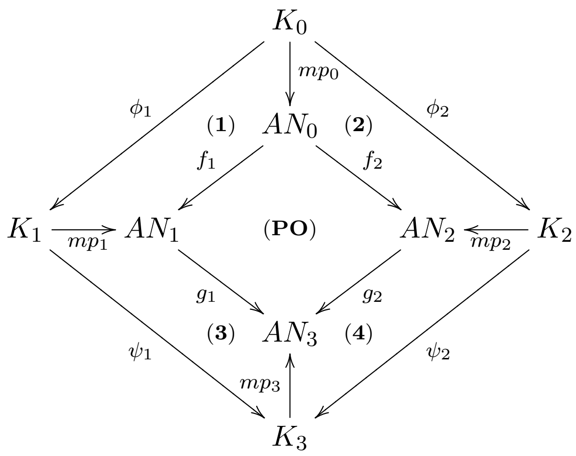
   establishing a bijective correspondence between $Proc(AN)$ and $Proc(AN_1 \overset{f_1}{\leftarrow} AN_0 \overset{f_2}{\rightarrow} AN_2)$.

Proof.

1. **Composition Construction.** The fact that (mp_0, ϕ_1) and (mp_0, ϕ_2) are agreement restrictions for mp_1 and mp_2 implies that (K_1, K_2) are composable w.r.t. (K_0, ϕ_1, ϕ_2) which by Lemma 5.2 implies that the composition $K_3 = K_1 +_{(K_0, \phi_1, \phi_2)} K_2$ exists and is an AHL-occurrence net. Since by Lemma 5.1 pushouts in **AHLONets** are also pushouts in **AHLNets** and there is

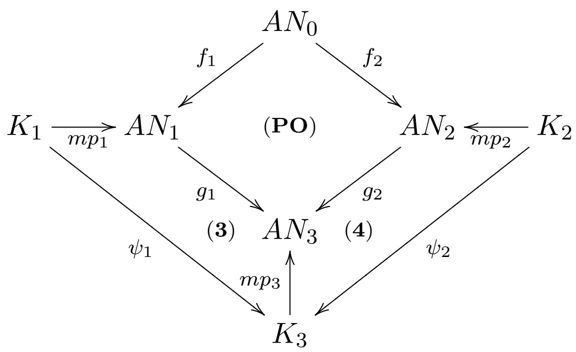
   $$g_1 \circ mp_1 \circ \phi_1 = g_2 \circ mp_2 \circ \phi_2$$

   the pushout property implies a unique AHL-morphism $mp_3 : K_3 \to AN_3$ such that (3) and (4) below commute.



From the fact that f_1, f_2, g_1 and g_2 are injective it follows that the above diagram is a weak Van Kampen cube with pushouts as top and bottom faces and pullbacks as back faces. The Van Kampen property implies that (3) and (4) are pullbacks and hence mp_3 is the amalgamation $mp_1 +_{m_0} mp_2$.

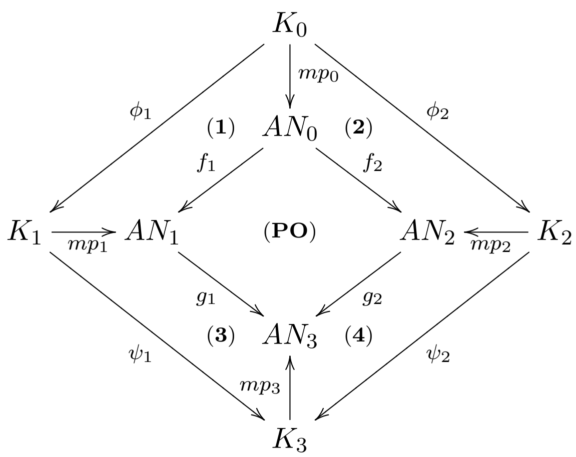
2. **Decomposition Construction.** Given restrictions (mp_1, ψ_1) and (mp_2, ψ_2) we have pullbacks (3) and (4) below in **AHLNets**.



Then we obtain the restriction (mp_0, ϕ_1) of mp_1 along f_1 as pullback (1) below in **AHLNets**. Furthermore there is

$$g_2 \circ f_2 \circ mp_0 = g_1 \circ f_1 \circ mp_0 = g_1 \circ mp_1 \circ \phi_1 = mp_3 \circ \psi_1 \circ \phi_1$$

which by the pullback property of (2) implies that there is a unique AHL-morphism $\phi_2 : K_0 \to K_2$ such that diagram (2) and the outer square below commute.



By composition of pullbacks we have that (1)+(3) is a pullback. Since (PO) and the outer square commute there is (1)+(3) = (2)+(4) which implies that (2)+(4) is a pullback and hence by pullback decomposition (2) is a pullback.
So we have that the above cube is a weak Van Kampen cube where all side faces are pullbacks and the bottom is a pushout. Hence the Van Kampen property implies that the top face (i. e. the outer square) is a pushout in **AHLNets**.
Now, by Lemma 3.1 the nets K_0, K_1 and K_2 are AHL-occurrence nets. This implies that the outer square which is a pushout in **AHLNets** is also a pushout in the full subcategory **AHLONets** $\subseteq$ **AHLNets** which by Fact 5.2 implies that (K_1, K_2) are composable w.r.t. (K_0, ϕ_1, ϕ_2). Hence, (mp_0, ϕ_1) and (mp_0, ϕ_2) are agreement restrictions for mp_1 and mp_2 which means that mp_3 is an amalgamation of mp_1 and mp_2.

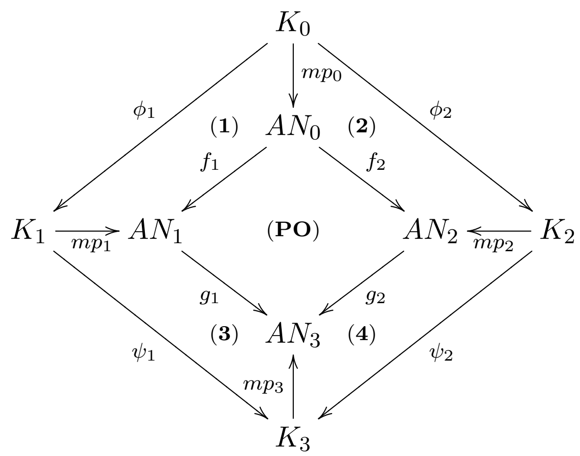
3. **Bijective Correspondence.** We define

$$Comp(\left[mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2\right]) = [mp_3]$$

where the AHL-occurrence net K_3 is obtained as composition of AHL-occurrence nets $K_3 = K_1 +_{(K_0,\phi_1,\phi_2)} K_2$ and the morphism $mp_3 : K_3 \to AN_3$ is the unique morphism induced by the pushout property of the corresponding pushout in **AHLNets**. Hence, $mp_3 = mp_1 \circ_{\phi_1,\phi_2} mp_2$ is unique up to isomorphism which means that the function *Comp* is well-defined.
Moreover, we define

$$Decomp([mp_3]) = \left[mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2\right]$$

where $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2$ is the amalgamation decomposition of mp_3 constructed as given in Item 2. The amalgamation decomposition constructed via pullbacks (1)-(4) in **AHLNets** is unique up to isomorphism due to the uniqueness of pullbacks. Thus, the function *Decomp* is well-defined.



Given an agreeing span $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2$ with respect to pushout (PO). Then by definition of agreement restrictions diagrams (1) and (2) above are pullbacks in **AHLNets**. The composition $mp_3 : K_3 \to AN_3$ is constructed via the pushout which is the outer square in the diagram above. Then the pushout (PO) is a weak Van Kampen square implying that (3) and (4) are pullbacks in **AHLNets**.
Since the decomposition of mp_3 is constructed via pullback (1)-(4) and pullbacks are unique up to isomorphism the result is isomorphic to $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2$, i. e.

$$Decomp(Comp(\left[mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2\right])) = \left[mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\rightarrow} mp_2\right]$$

Vice versa, given an AHL-process $mp_3 : K_3 \to AN_3$ the amalgamation decomposition $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\to} mp_2$ of mp_3 is constructed via pullbacks (1)-(4) leading to the fact that (PO) is a weak Van Kampen square. This implies that the outer square is a pushout which defines exactly the composition of $mp_1 \overset{\phi_1}{\leftarrow} mp_0 \overset{\phi_2}{\to} mp_2$. Since pushouts are unique up to isomorphism there is

$$Comp(Decomp([mp_3])) = [mp_3]$$

Hence, *Comp* and *Decomp* are inverse to each other which means that they are bijections.

□

References

[BCEH01] P. Baldan, A. Corradini, H. Ehrig, and R. Heckel. Compositional modeling of reactive systems using open nets. In *CONCUR '01: Proceedings of the 12th International Conference on Concurrency Theory*, pages 502–518, London, UK, 2001. Springer.

[Bro85] M. Broy. Specification and top down design of distributed systems. In *Proc. TAPSOFT'85*, volume 1 of *LNCS 185*, 1985.

[CIP81] CIP Language Group. Report on a wide spectrum language for program specification and development. Technical Report TUM-I8104, TU München, 1981. also available as Springer LNCS 183.

[EEPT06] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation.* EATCS Monographs in TCS. Springer, 2006.

[EHGP09] H. Ehrig, K. Hoffmann, K. Gabriel, and J. Padberg. Composition and independence of high-level net processes. *Electronic Notes in Theoretical Computer Science*, 242(2):59 – 71, 2009. Proceedings of the First Workshop on Formal Methods for Wireless Systems (FMWS 2008).

[EHP+02] H. Ehrig, K. Hoffmann, J. Padberg, P. Baldan, and R. Heckel. High-level net processes. In *Formal and Natural Computing*, volume 2300 of *LNCS*, pages 191–219. Springer, 2002.

[Ehr79] H. Ehrig. Introduction to the algebraic theory of graph grammars (a survey). In V. Claus, H. Ehrig, and G. Rozenberg, editors, *Graph Grammars and Their Application to Computer Science and Biology, Lecture Notes in Computer Science,* No. 73, pages 1–69. Springer, 1979.

[Ehr05] H. Ehrig. Behaviour and Instantiation of High-Level Petri Net Processes. *Fundamenta Informaticae*, 65(3):211–247, 2005.

[EKT+80] H. Ehrig, H.-J. Kreowski, J.W. Thatcher, E.G. Wagner, and J.B. Wright. Parametrized data types in algebraic specification languages. In *7th Int. Coll. Automata, Languages, and Programming*, Springer LNCS 85, pages 157–168, 1980.

[ER76] H. Ehrig and B.K. Rosen. Commutativity of independent transformations on complex objects. Technical Report RC 6251, IBM Research, 1976.

[ER97] Hartmut Ehrig and Wolfgang Reisig. An Algebraic View on Petri Nets. *Bulletin of the EATCS*, 61:52–58, February 1997.

[Goo10a] Google. *http://google.com*, September 2010.

[Goo10b] Google Wave. *https://wave.google.com/wave*, September 2010.

[GR83] U. Goltz and W. Reisig. The Non-sequential Behavior of Petri Nets. *Information and Control*, 57(2/3):125–147, 1983.

[GTWW75] J.A. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright. Abstract data types as initial algebras and the correctness of data representations. In *Proc. of Conf. on Computer Graphics*, 1975.

[Gut75] J.V. Guttag. *The specification and application to programming of abstract data types.* PhD thesis, University of Toronto, 1975.

[HKR80] U.L Hupbach, H. Kaphengst, and H. Reichel. Initial algebraic specification of data types, parameterized data types and algorithms. Technical Report 15, VEB Robotron ZFT, Dresden, 1980.

[Jen91] K. Jensen. Coloured petri nets: A high-level language for system design and analysis. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *LNCS*, pages 342–416. Springer, 1991.

[MEE+10] Tony Modica, Claudia Ermel, Hartmut Ehrig, Kathrin Hoffmann, and Enrico Biermann. Modeling communication spaces with higher-order petri nets. In George E. Lasker and Jochen Pfalzgraf, editors, *Advances in Multiagent Systems, Robotics and Cybernetics: Theory and Practice*, volume III, Tecumseh, Canada, 2010. The International Institute for Advanced Studies in Systems Research and Cybernetics.

[MGE+10] Tony Modica, Karsten Gabriel, Hartmut Ehrig, Kathrin Hoffmann, Sarkaft Shareef, Claudia Ermel, Ulrike Golas, Frank Hermann, and Enrico Biermann. Low- and High-Level Petri Nets with Individual Tokens. Technical Report 2009/13, Technische Universität Berlin, 2010. `http://www.eecs.tu-berlin.de/menue/forschung/forschungsberichte/2009`.

[Mil80] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[MM90] J. Meseguer and U. Montanari. Petri Nets Are Monoids. *Information and Computation*, 88(2):105–155, 1990.

[PER95] J. Padberg, H. Ehrig, and L. Ribeiro. Algebraic high-level net transformation systems. *Mathematical Structures in Computer Science*, 80:217–259, 1995.

[Pet62] C.A. Petri. *Kommunikation mit Automaten.* PhD thesis, Institut für instrumentelle Mathematik, Universität Bonn, 1962.

[Rei85] W. Reisig. *Petrinetze, Eine Einführung.* Springer Verlag, Berlin, 1985.

[Rei90] W. Reisig. Petri nets and algebraic specifications. Technische Universität München, SFB-Bericht 342/1/90 B, March, 1990.

[Roz87] G. Rozenberg. Behaviour of Elementary Net Systems. In *Petri Nets: Central Models and Their Properties, Advances in Petri Nets*, volume 254 of *LNCS*, pages 60–94. Springer, 1987.

[Roz97] G. Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformation, Vol 1:* Foundations. World Scientific, Singapore, 1997.

[Yon10] Tsvetelina Yonova. Formal description and analysis of distributed online collaboration platforms. Bachelor thesis, Technische Universität Berlin, 2010.

[Zil74] S.N. Zilles. Algebraic specification of data types. *Project MAC Progress Report 11, MIT*, pages 28–52, 1974.