Preserving Behavior in Transition Systems from Event Structure Models*

Nataliya Gribovskaya[1] and Irina Virbitskaite[1,2]

[1] A.P. Ershov Institute of Informatics Systems, SB RAS,
6, Acad. Lavrentiev av., 630090 Novosibirsk, Russia
[2] Novosibirsk State University, 2, Pirogov av., 630090 Novosibirsk, Russia
{gribovskaya,virb}@iis.nsk.su

Abstract. Two structurally different methods of associating transition system semantics to event structure models are distinguished in the literature. One of them is based on configurations (event sets), the other on residuals (model fragments). In this paper, we consider three kinds of event structures (resolvable conflict structures, extended prime structures, stable structures), translate the other models into resolvable conflict structures and back, provide the isomorphism results on the two types of transition systems, and demonstrate the preservation of some bisimulations on them.

1 Introduction

Since the introduction of event structures in [26], many variants of event-oriented models have been proposed based on different behavioural relations between events and thus providing a different expressive power. Among the models are prime event structures [26] (with conjunctive[1] binary causality, represented by a partial order and being under the principle of finite causes, and symmetric irreflexive conflict, obeying the principle of conflict heredity; all these guarantee unique event enablings within the model); extended prime event structures [1] (with conjuctive binary causality, being possibly with cycles and not being under the principle of finite causes, and symmetric irreflexive conflict, not obeying the principle of conflict heredity; moreover, the relations can be overlapped); stable event structures [28] (with non-binary conjuctive causality, allowing for alternative enablings, and the stability constraint (i.e .the intersection of two non-conflicting causal predecessors sets for an event is a causal predecessors set for the event) resulting in unique enablings within a configuration); event structures for resolvable conflict [14] (with dynamic conflicts, i.e. conflicts can be resolved or created by the occurrences of other events), etc. Comparative analysis of some classes of event structures can be found in the works [1, 2, 11, 12, 14, 15, 16, 18].

Two methods of associating a labeled transition system [20] with an event-oriented model of a distributed system, such as an event structure [26] or a

* Supported by German Research Foundation through grant Be 1267/14-1.
[1] An event is enabled once all of its causal predecessors have occurred.

configuration structure [13], can be distinguished: a *configuration-based* and a *residual-based* method. In the first case,[2] states are understood as sets of events, called *configurations*, and state transitions are built by starting with the empty configuration and enlarging configurations by already executed events. In the second approach,[3] states are understood as event structures, and transitions are built by starting with the given event structure as an initial state and removing already executed parts thereof in the course of an execution.

In the literature, configuration-based transition systems seem to be predominantly used as the semantics of event structures, whereas residual-based transition systems are actively used in providing operational semantics of process calculi and in demonstrating the consistency of operational and denotational semantics. The two kinds of transition systems have occasionally been used alongside each other (see [18] as an example), but their general relationship has not been studied for a wide range of existing models. In a seminal paper, viz. [23], bisimulations between configuration-based and residual-based transition systems have been proved to exist for prime event structures [28]. The result of [23] has been extended in [5] to more complex event structure models with asymmetric conflict. Counterexamples illustrated that an isomorphism cannot be achieved with the various removal operators defined in [5, 23]. The paper [6] demonstrated that the operators can be tightened in such a way that *isomorphisms*, rather than just bisimulations, between the two types of transition systems belonging to a single event structure can be obtained. A key idea is to employ *non-executable (impossible)* events[4] if the model allows them (and to introduce a special non-executable event otherwise), in order to turn *model fragments* into parts of states. This idea has been applied by the authors on a wide variety of event structure models, and for a full spectrum of semantics (interleaving, step, pomset, multiset). Thanks to the results, a variety of facts known from the literature on configuration-based transition systems (e.g., [4, 10, 13, 28]) can be extended to residual-based ones.

The aim of this paper is to connect several models of event structures by providing behaviour preserving translations between them, and to demonstrate the retention of some of the bisimulation concepts in the two types of transition systems associated with the models under consideration.

In Section 2 of this paper, we consider three kinds of event structure models (resolvable conflict, extended prime, stable event structures), define removal operators for them, and translate the other models into resolvable conflict event structures and back. Section 3 contains the definitions of the two types of transition systems, describes the isomorphism results, and demonstrate the preservation of some bisimulations on the transition systems. Section 4 concludes.

[2] E.g., see [1, 2, 11, 12, 14, 15, 17, 18, 24, 27].

[3] E.g., see [3, 7, 8, 9, 18, 19, 21, 22, 25].

[4] In an event structure, an event is called non-executable or impossible if it does not occur in any configuration of the structure, i.e. the event is never executed.

2 Event Structure Models

2.1 Event Structures for Resolvable Conflict

In this section, we consider event structures for resolvable conflict, which were put forward in [14] to give semantics to general Petri nets. A structure for resolvable conflict consists of a set of events and an enabling relation $\vdash$ between sets of events. The enabling $X \vdash Y$ with sets X and Y imposes restrictions on the occurrences of events in Y by requiring that for all events in Y to occur, their *causes* – the events in X – have to occur before. This allows for modeling the case when a and b cannot occur together until c occurs, i.e., initially a and b are in conflict until the occurrence of c resolves this conflict. Notice that the event structure classes under consideration in this paper are unable to model the phenomena of resolvable conflict. In resolvable conflict structures, the enabling relation can also model conflicts: events from a set Y are *in irresolvable conflict* iff there is no enabling of the form $X \vdash Y$ for any set X of events. Further, an event can be impossible (i.e. non-executable in any system's run) if it has no enabling or has infinite causes or has impossible causes/prececessors.

Definition 1. *An* event structure for resolvable conflict (RC-structure) over L *is a tuple* $\mathcal{E} = (E, \vdash, L, l)$*, where* E *is a set of events;* $\vdash \subseteq \mathcal{P}(E) \times \mathcal{P}(E)$ *is the* enabling relation*;* L *is a set of labels;* $l : E \to L$ *is a labeling function.*

Let $\mathcal{E}$ be an RC-structure over L. For $X \subseteq E$ and $e \in E$, $Con(X) \iff \forall \widehat{X} \subseteq X\colon \exists Z \subseteq E\colon Z \vdash \widehat{X}$; $fCon(X) \iff X$ is finite and $Con(X)$. The *conflict relation* $\sharp \subseteq E \times E$ is given by: $d \sharp e \iff d \neq e \wedge \neg Con(\{d, e\})$. The *direct causality relation* $\prec \subseteq E \times E$ is defined as follows: $d \prec e \iff \forall X \subseteq E\colon (X \vdash e \Rightarrow d \in X)$. A set $X \subseteq E$ is *left-closed* iff X is finite, and for all $\widetilde{X} \subseteq X$ there exists $\widehat{X} \subseteq X$ such that $\widehat{X} \vdash \widetilde{X}$. The set of the left-closed sets of $\mathcal{E}$ is denoted as $LC(\mathcal{E})$. Clearly, any left-closed set is conflict-free. Let $Conf(\mathcal{E}) = \{\{e_1, \ldots, e_n\} \subseteq E \mid n \geq 0,\ \forall i \leq n\colon \forall X \subseteq \{e_1, \ldots, e_i\}\colon \exists Y \subseteq \{e_1, \ldots, e_{i-1}\}\colon Y \vdash X\}$ be the set of configurations of $\mathcal{E}$. Clearly, any configuration X is a left-closed set but not conversely.

Consider some properties of resolvable conflict event structures.

Definition 2. *An* RC*-structure* $\mathcal{E} = (E, \vdash, L, l)$ *is called*

- rooted *iff* $(\emptyset, \emptyset) \in \vdash$*;*
- pure *iff* $X \vdash Y \Rightarrow X \cap Y = \emptyset$*;*
- singular *iff* $X \vdash Y \Rightarrow X = \emptyset \vee \mid Y \mid = 1$*;*
- manifestly conjunctive *iff there is at most one* X *with* $X \vdash Y$*, for all* $Y \subseteq E$*;*
- conjuctive *iff* $X_i \vdash Y (i \in I \neq \emptyset) \Rightarrow \bigcap_{i \in I} X_i \vdash Y$*;*
- locally conjuctive *iff* $X_i \vdash Y (i \in I \neq \emptyset) \wedge Con(\bigcup_{i \in I} X_i \cup Y) \Rightarrow \bigcap_{i \in I} X_i \vdash Y$*;*
- with finite causes *iff* $X \vdash Y \Rightarrow X is finite$*;*
- with binary conflict *iff* $\mid X \mid > 2 \Rightarrow \emptyset \vdash X$*;*
- in the standard form *iff* $\vdash = \{(A, B) \mid A \cap B = \emptyset,\ A \cup B \in LC(\mathcal{E})\}$*;*
- 2-coherent *iff* $X \cup Y \in LC(\mathcal{E})$*, for all* $X, Y \in LC(\mathcal{E})$ *s.t.* $X \cup Y \subseteq Z \in LC(\mathcal{E})$.[5]

[5] This property is useful when proving Theorem 1.

Lemma 1. *An RC-structure $\mathcal{E} = (E, \vdash, L, l)$ can be transformed into:*

- *a pure RC-structure $PU(\mathcal{E}) = (E, \widehat{\vdash}, L, l)$*[6] *s.t. $Conf(\mathcal{E}) = Conf(PU(\mathcal{E}))$, if $\mathcal{E}$ is a singular RC-structure;*
- *an RC-structure $SF(\mathcal{E}) = (E, \widetilde{\vdash}, L, l)$*[7] *in the standard form s.t. $LC(\mathcal{E}) = LC(SF(\mathcal{E}))$. Moreover, $Conf(\mathcal{E}) = Conf(SF(\mathcal{E}))$, if $\mathcal{E}$ is a pure RC-structure.*

Example 1. As an example, consider the pure, manifestly conjuctive, non-singular, non-2-coherent *RC*-structure $\mathcal{E}^{rc} = (E^{rc}, \vdash^{rc}, L, l^{rc})$ with finite causes and binary conflict from [15], where $E^{rc} = \{a, b, c\}$; $\vdash^{rc}$ consists of $\emptyset \vdash X$ for all $X \neq \{a, b\}$ and $\{c\} \vdash \{a, b\}$; $L = E^{rc}$; and l^{rc} is the identity labeling function. It is easy to see that $LC(\mathcal{E}^{rc}) = Conf(\mathcal{E}^{rc}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$. This *RC*-structure models the initial conflict between the events a and b that can be resolved by the occurrence of the event c. The structure $\mathcal{E}^{rc}$ can be presented in the standard form $\widetilde{\mathcal{E}^{rc}}$ with $\widetilde{\vdash}^{rc}$ consisting of $A \widetilde{\vdash} B$ such that $B \subseteq C \in LC(\mathcal{E})$ and $A = C \setminus B$, i.e. $\widetilde{\vdash}^{rc} = \{(\emptyset, \emptyset), (\emptyset, \{a\}), (\{a\}, \emptyset), (\emptyset, \{b\}), (\{b\}, \emptyset), (\emptyset, \{c\}), (\{c\}, \emptyset), (\emptyset, \{a, c\}), (\{a, c\}, \emptyset), (\{a\}, \{c\}), (\{c\}, \{a\}), \ldots, (\emptyset, \{a, b, c\}), (\{a, b, c\}, \emptyset)\}$.

The standard form of *RC*-structures and the ability to specify impossible events in the model allows for developing a relatively simple structural definition of a removal operator which is necessary for residual semantics.

Definition 3. *For an RC-structure $\mathcal{E} = (E, \vdash, L, l)$ in the standard form and $X \in LC(\mathcal{E})$, a removal operator is defined as follows: $\mathcal{E} \setminus X = (E', \vdash', L, l')$, where*

$$E' = E \setminus X$$
$$\vdash' = \{(A', B') \mid \exists (A, B) \in \vdash \text{ s.t. } A' = A \cap E', B' = B \cap E', (A' \cup B' \cup X) \in LC(\mathcal{E})\}$$
$$l' = l\,|_{E'}$$

According to the definition above, all the events in X are removed; however, we retain the events, not forming left-closed sets with the events in X and hence conflicting with some events in X, making the retained events impossible by deleting their enabling relations.

From now on, we use $\mathbb{E}_L^{rc}$ to denote the class of rooted, singular, locally conjuctive *RC*-structures with binary conflict.

2.2 Extended Prime Event Structures

For reasons of flexibility, the authors of [1] propose to generalise ordinary prime event structures [28][8] by dropping the transitivity and acyclicity of causality,

[6] An *RC*-structure $PU(\mathcal{E}) = (E, \widehat{\vdash}, L, l)$ can be directly obtained by putting $\widehat{\vdash} = \vdash \setminus \{(A, B) \in \vdash \mid \emptyset \neq B \subseteq A\}$.

[7] An *RC*-structure $SF(\mathcal{E}) = (E, \widetilde{\vdash}, L, l)$ can be directly obtained by putting $\widetilde{\vdash} = \{(A, B) \mid B \subseteq C \in LC(\mathcal{E}), A = C \setminus B\}$.

[8] A labeled prime event structure over a set L of actions is a tuple $\mathcal{E} = (E, \sharp, \leq, L, l)$, where E is a set of events; $\leq \subseteq E \times E$ is a partial order (the *causality relation*),

as well as the principles of finite causes and conflict inheritance.[9] As opposed to prime event structures, the extended version allows for impossible events. In this model, events can be impossible because of enabling cycles, or an overlapping between the enabling and the conflict relation, or because of impossible causes/predecessors.

Definition 4. *An* extended prime event structure (EP-structure) over L *is a triple* $\mathcal{E} = (E, \sharp, \prec, L, l)$*, where* E *is a set of events;* $\sharp \subseteq E \times E$ *is an irreflexive symmetric relation (the* conflict relation*);* $\prec \subseteq E \times E$ *is the* enabling relation*;* L *is a set of labels;* $l : E \to L$ *is a labeling function. Let* $\mathbb{E}_L^{ep}$ *denote the class of* EP*-structures over* L*.*

Let $\mathcal{E} = (E, \sharp, \prec, L, l)$ be an EP-structure. For $e \in E$, define $\downarrow e$ as a maximal subset of E such that $\forall e' \in \downarrow e \colon e' \prec e$. For $X \subseteq E$, let $\sharp(X) = \{e' \in E \mid \exists e \in X \colon e \sharp e'\}$. We call a set $X \subseteq E$ a *configuration* of $\mathcal{E}$ if X is finite, conflict-free (i.e. $\forall e, e' \in X \colon \neg(e \sharp e')$), left-closed (i.e. $\forall e, e' \in E \colon e \prec e' \wedge e' \in X \Rightarrow e \in X$), and does not contain enabling cycles (i.e., $\nexists e_1, \dots, e_n \in X \colon e_1 \prec \dots \prec e_n \prec e_1$ $(n \geq 1)$). The set of configurations of $\mathcal{E}$ is denoted by $Conf(\mathcal{E})$.

In the graphical representation of an EP-structure, pairs of events related by the enabling relation are connected by arrows; pairs of the events included in the conflict relation are marked by the symbol $\sharp$.

$\mathcal{E}^{ep}$:

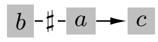

Fig. 1. An extended prime event structure $\mathcal{E}^{ep}$

Example 2. Figure 1 depicts the EP-structure $\mathcal{E}^{ep}$ over $L = \{a, b, c\}$, with $E^{ep} = L$; $\sharp^{ep} = \{(a, b),\ (b, a)\}$; $\prec^{ep} = \{(a, c)\}$; and the identity labeling function l^{ep}. Observe that the principle of conflict inheritance is violated. The set of configurations of $\mathcal{E}^{ep}$ is $\{\emptyset,\ \{a\},\ \{b\},\ \{a, c\}\}$.

Consider the definition of the removal operator for EP-structures.

Definition 5. *For* $\mathcal{E} \in \mathbb{E}_L^{ep}$ *and* $X \in Conf(\mathcal{E})$*, a removal operator is defined as follows:* $\mathcal{E} \setminus X = (E',\ \prec',\ \sharp',\ L,\ l')$*, with*

$$\begin{aligned} E' &= E \setminus X \\ \sharp' &= \sharp \cap (E' \times E') \\ \prec' &= (\prec \cap (E' \times E')) \ \cup\ \{(e, e) \mid e \in \sharp(X)\} \\ l' &= l \mid_{E'} \end{aligned}$$

satisfying the *principle of finite causes*: $\forall e \in E \colon \lfloor e \rfloor = \{e' \in E \mid e' \leq e\}$ is finite; $\sharp \subseteq E \times E$ is an irreflexive and symmetric relation (the *conflict relation*), satisfying the *principle of hereditary conflict*: $\forall e, e', e'' \in E \colon e \leq e'$ and $e \sharp e''$ then $e' \sharp e''$; and $l : E \to L$ is a labeling function.

[9] It was noted in [1] that, as far as finite configurations are concerned, this does not lead to an increase in expressive power.

We see that the events in X are removed, yielding a reduction of the enabling and conflict relations. At the same time, any event conflicting with some event in X is retained, equipping it with an enabling cycle, thereby making the conflicting event impossible.

Translate EP-structures into RC-structures and conversely. For an EP-structure $EP = (E, \sharp, \prec, L, l)$, define $\mathcal{RC}(EP) = (E' = E, \vdash', L, l = l')$, where

$$X \vdash' Y \iff \begin{cases} \text{either } Y = \{e\}, X = \downarrow e, \\ \text{or } Y = \{e, e'\}, e \neq e', \neg(e \sharp e'), X = \emptyset, \\ \text{or } \mid Y \mid \neq 1, 2, X = \emptyset. \end{cases}$$

For an RC-structure $RC = (E', \vdash', L, l')$, let $\mathcal{EP}(RC) = (E'' = E', \sharp'' = \sharp', \prec'' = \prec', L, l'' = l')$.

Lemma 2. *(i) For EP an EP-structure, $\mathcal{RC}(EP)$ is a rooted, singular, manifestly conjuctive RC-structure with binary conflict such that $Conf(EP) = Conf(\mathcal{RC}(EP))$.*

(ii) For RC a rooted, singular, conjuctive RC-structure with binary conflict, $\mathcal{EP}(RC)$ is an EP-structure such that $Conf(RC) = Conf(\mathcal{EP}(RC))$.

2.3 Stable Event Structures

Stable event structures, introduced in the work of Winskel [27] in order to overcome the unique enabling problem of prime event structures, have an enabling relation indicating which (usually finite) sets X of events are possible prerequisites of a single event e, written $X \vdash e$. We consider the version of stable event structures of [28] where the conflict relation is specified for sets with two events.

Definition 6. *A* stable event structure over L (S-structure) *is a tuple* $\mathcal{E} = (E, \sharp, \vdash, L, l)$*, where*

- *E is a set of events;*
- *$\sharp \subseteq E \times E$ is an irreflexive, symmetric relation (the* conflict relation*). We shall write Con for the set of finite conflict-free subsets of E, i.e. those finite subsets $X \subseteq E$ for which $\forall e, e' \in X : \neg(e \sharp e')$. $X \in Con$ means that the events in X could happen in the same run, i.e. they are consistent;*
- *$\vdash \subseteq Con \times E$ is the* enabling relation *which satisfies $X \vdash e$ and $X \subseteq Y \in Con \Rightarrow Y \vdash e$; and, moreover, $X \vdash e$, $Y \vdash e$, and $X \cup Y \cup \{e\} \in Con \Rightarrow X \cap Y \vdash e$ (*the stability principle*). $\vdash$ indicates possible causes: an event e can occur whenever for some X with $X \vdash e$ all events in X have occurred before. The minimal enabling relation $\vdash_{min}$ is defined as follows: $X \vdash_{min} e$ iff $X \vdash e$ and for all $Y \subseteq X$ if $Y \vdash e$ then $Y = X$;*
- *L is a set of actions;*
- *$l : E \to L$ is a labeling function.*

Let $\mathbb{E}^s_L$ denote the class of S-structures over L.

A set $X \subseteq E$ is a *configuration* of an S-structure $\mathcal{E}$ iff X is finite, conflict-free (i.e., $X \in Con$), and secured (i.e., there are $e_1, \ldots, e_n$ such that $X = \{e_1, \ldots, e_n\}$

and $\{e_1, \ldots, e_i\} \vdash e_{i+1}$, for all $i < n$). The set of configurations of $\mathcal{E}$ is denoted $Conf(\mathcal{E})$. For an S-structure $\mathcal{E}$, $X \in Conf(\mathcal{E})$, and $e, e' \in X$, let $e' \prec_X e$ iff e' belongs to the smallest subset Y of X with $Y \vdash e$.

Example 3. Consider the S-structure $\mathcal{E}^s$ over $L = \{a, b, c, d\}$, with $E^s = L$; $\sharp^s = \{(a, b), (b, a)\}$; $\vdash^s_{min} = \{(\emptyset, a), (\emptyset, b), (\emptyset, c), (\{a\}, d), (\{b, c\}, d)\}$; and the identity labeling function l^s. The set of configurations of $\mathcal{E}^s$ is $\{\emptyset, \{a\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{a, d\}, \{a, c, d\}, \{b, c, d\}\}$. Notice that $\mathcal{E}^s$ is not a flow event structure because the event c not conflicting with the event a may be a cause for d or may not.

Definition 7. *For $\mathcal{E} = (E, \sharp, \vdash, L, l) \in \mathbb{E}^s_L$ and $X \in Conf(\mathcal{E})$, a removal operator is defined as follows: $\mathcal{E} \setminus X = (E', \sharp', \vdash', L, l')$, with*

$$
\begin{aligned}
E' &= E \setminus X \\
\sharp' &= \sharp \cap (E' \times E') \\
\vdash' &= \{(W', e) \mid W' \in Con', \ \exists (W'', e) \in \vdash'_{min} \ \ s.t. \ W'' \subseteq W'\} \ where \\
&\quad \vdash'_{min} = \{(W'', e) \mid \exists (W, e) \in \vdash_{min} \ \ s.t. \ W'' = W \cap E', \ e \in E', \\
&\qquad\qquad\qquad\qquad\qquad\qquad W'' \cup X \in Con, \ \{e\} \cup X \in Con\} \\
l' &= l\,|_{E'}
\end{aligned}
$$

We see that all the events in X are deleted; the conflict relation $\sharp'$ contains the pairs of remaining conflicting events; the definition of $\vdash'$ is based on that of $\vdash'_{min}$, which consists of the pairs from $\vdash$ without the pairs whose events conflict with some event in X, thereby making them impossible.

For an S-structure $S = (E, \sharp, \vdash, L, l)$, let $\mathcal{RC}(G) = (E' = E, \vdash', L, l')$, where

$$
X \vdash' Y \iff \begin{cases} \text{either } Y = \{e\}, e \in E, X \vdash e, \\ \text{or } \ | Y | = 2, Y \in Con, X = \emptyset, \\ \text{or } \ | Y | \neq 1, 2, X = \emptyset. \end{cases}
$$

For an RC-structure $RC = (E', \vdash', L, l')$, let $\mathcal{S}(RC) = (E'' = E', \sharp'' = \sharp', \vdash'', L, l'')$, where $X \vdash'' e \iff e \in E'$, $X \subseteq E'$, $fCon'(X)$, and $\exists Y \subseteq X : Y \vdash' \{e\}$.

Lemma 3. *[15]*

(i) For S an S-structure, $\mathcal{RC}(S)$ is a rooted, singular, locally conjuctive RC-structure with finite causes and binary conflict s.t. $Conf(S) = Conf(\mathcal{RC}(F))$.
(ii) For RC a rooted, singular, locally conjuctive RC-structure with finite causes and binary conflict, $\mathcal{S}(RC)$ is an S-structure s.t. $Conf(RC) = Conf(\mathcal{S}(RC))$.

2.4 Different Semantics

In this subsection, we define different semantics for the event structure models under consideration. From now on, we treat $\mathcal{E}$ as an event structure over L specified in Definitions 1, 4, and 6, if not defined otherwise. Moreover, let $\mathbb{E}_L = \mathbb{E}^{ep}_L \cup \mathbb{E}^s_L \cup \mathbb{E}^{rc}_L$.

We first introduce auxiliary notations. Given configurations $X, X' \in Conf(\mathcal{E})$, we write:

- $X \rightarrow_{int} X'$ iff $X \subseteq X'$ and $X' \setminus X = \{e\}$;
- $X \rightarrow_{step} X'$ iff $X \subseteq X'$ and $X'' \in \mathit{Conf}(\mathcal{E})$, for all $X \subseteq X'' \subseteq X'$;
- $X \rightarrow_{pom} X'$ iff $X \subseteq X'$ and $\leq_{X' \setminus X}$ is a partial order;
- $X \rightarrow_{whp} X'$ iff $X \subseteq X'$ and $\leq_{X'}$ is a partial order.

For $\star \in \{int, step, pom\}$, a configuration $X \in \mathit{Conf}(\mathcal{E})$ is a *configuration in $\star$-semantics* of $\mathcal{E}$ iff $\emptyset \rightarrow_\star^* X$, where $\rightarrow_\star^*$ is the reflexive and transitive closure of $\rightarrow_\star$. Let $\mathit{Conf}_\star(\mathcal{E})$ denote the set of configurations in $\star$-semantics of $\mathcal{E}$.

Lemma 4. *Given an event structure $\mathcal{E} \in \mathbb{E}_L$ and $\star, * \in \{int, step, pom\}$,*

(i) for a configuration $X \in \mathit{Conf}(\mathcal{E})$, the transitive and reflexive closure of $\prec_X$, $\leq_X$, is a partial order. Let $\mathcal{E}\lceil X = (X, \leq_X, L, l\mid_X)$;
(ii) $\mathit{Conf}(\mathcal{E}) = \mathit{Conf}_\star(\mathcal{E}) = \mathit{Conf}_*(\mathcal{E})$.

Given $\star \in \{int, step, pom\}$, an event structure $\mathcal{E}$ over L, and configurations $X, X' \in \mathit{Conf}_\star(\mathcal{E})$ such that $X \rightarrow_\star X'$, we write:

- $l_{int}(X' \setminus X) = a$ iff $X' \setminus X = \{e\}$ and $l(e) = a$, if $\star = int$;
- $l_{step}(X' \setminus X) = M$ iff $M(a) = |\{e \in X' \setminus X \mid l(e) = a\}|$, for all $a \in L$, if $\star = step$;
- $l_{pom}(X' \setminus X)=\mathcal{Y}$ iff $\mathcal{Y} = [(X' \setminus X, \leq_{X'} \cap (X' \setminus X \times X' \setminus X), L, l\mid_{X' \setminus X})]$, if $\star = pom$;
- $l_{whp}(X)=\mathcal{Y}$ iff $\mathcal{Y} = [(X, \leq_X, L, l\mid_X)]$.

Let $\mathcal{E}$ be an event structure over L and $X = \{e_1, \ldots, e_n\} \in \mathit{Conf}_{int}(\mathcal{E})$ ($n \geq 0$). We call $e_1 \ldots e_n$ a *derivation* of X iff $X_0 = \emptyset \rightarrow_{int} X_1 \ldots X_{n-1} \rightarrow_{int} X_n = X$, and $X_i \setminus X_{i-1} = \{e_i\}$, for all $1 \leq i \leq n$. A derivation $e_1 \ldots e_n$ of $X \in \mathit{Conf}_{int}(\mathcal{E})$ and a derivation $f_1 \ldots f_n$ of $X' \in \mathit{Conf}_{int}(\mathcal{E}')$ are *equal* (denoted $e_1 \ldots e_n \sim f_1 \ldots f_n$) iff there is an isomorphism $\iota : \mathcal{E}\lceil X \rightarrow \mathcal{E}'\lceil X'$ with $\iota(e_1 \ldots e_n) := \iota(e_1) \ldots \iota(e_n) = f_1 \ldots f_n$. Let $\mathit{Der}(X)$ denote the set of all equivalence classes $[e_1 \ldots e_n]$ of derivations of X. For $[e_1 \ldots e_n] \in \mathit{Der}(X)$, define $l_{hp}([e_1 \ldots e_n]) := a_1 \ldots a_n$, where $l_i(e_i) = a_i$ ($1 \leq i \leq n$).

3 Transition Systems $\mathit{TC}(\mathcal{E})$ and $\mathit{TR}(\mathcal{E})$

3.1 Definitions and Comparisons

In this subsection, we first give some basic definitions concerning labeled transition systems, and then define the mappings $\mathit{TC}(\mathcal{E})$ and $\mathit{TR}(\mathcal{E})$, which associate two distinct kinds of transition systems – one whose states are configurations and one whose states are residual event structures – with an event structure $\mathcal{E}$ over L.

A transition system $T = (S, \rightarrow, i)$ over a set $\mathcal{L}$ of labels consists of a set of states S, a transition relation $\rightarrow \subseteq S \times \mathcal{L} \times S$, and an initial state $i \in S$. Two transition systems over $\mathcal{L}$ are *isomorphic* if their states can be mapped one-to-one to each other, preserving transitions and initial states. We call a relation $R \subseteq S \times S'$ a *bisimulation* between transition systems T and T' over $\mathcal{L}$ iff

$(i,i') \in R$, and for all $(s,s') \in R$ and $l \in \mathcal{L}$: if $(s,l,s_1) \in \rightarrow$, then $(s',l,s_1') \in \rightarrow$ and $(s_1,s_1') \in R$, for some $s_1' \in S'$; and if $(s',l,s_1') \in \rightarrow$, then $(s,l,s_1) \in \rightarrow$ and $(s_1,s_1') \in R$, for some $s_1 \in S$.

Introduce additional auxiliary notations. For a fixed set L of labels of event structures, define $\mathbb{L}_{int} := L$, $\mathbb{L}_{pom} := Pom_L$ (the set of isomorphic classes of partial orders labeled over L), and $\mathbb{L}_{Der} := L^*$, being another sets of labels of the transition systems.

We are ready to define labeled transition systems with configurations as states.

Definition 8. *For an event structure $\mathcal{E}$ over L and $\star \in \{int, step, pom\}$,*

- *$TC_\star(\mathcal{E})$ is the transition system $(Conf_\star(\mathcal{E}), \rightharpoondown_\star, \emptyset)$ over $\mathbb{L}_\star$, where $X \overset{p}{\rightharpoondown}_\star X'$ iff $X \rightarrow_\star X'$ and $p = l_\star(X' \setminus X)$;*
- *$TC_{whp}(\mathcal{E})$ is the transition system $(Conf_{int}(\mathcal{E}), \rightharpoondown_{whp}, \emptyset)$ over $\mathbb{L}_{pom}$, where $X \overset{p}{\rightharpoondown}_{whp} X'$ iff $X \rightarrow_{whp} X'$ and $p = l_{whp}(X')$;*
- *$TC_{hp}(\mathcal{E})$ is the transition system $(\{Der(X) \mid X \in Conf_{int}(\mathcal{E})\}, \rightharpoondown_{hp}, \epsilon)$ over $\mathbb{L}_{Der}$, where $[e_1 \dots e_n] \overset{q}{\rightharpoondown}_{hp} [e_1 \dots e_n e_{n+1}]$ $(n \geq 0)$ iff $\{e_1, \dots, e_n\}$, $\{e_1, \dots, e_n, e_{n+1}\} \in Conf_{int}(\mathcal{E})$, and $q = l_{hp}([e_1 \dots e_n e_{n+1}])$.*

Consider the definition of labeled transition systems with residuals as states.

Definition 9. *For an event structure $\mathcal{E}$ over L and $\star \in \{int, step, pom\}$,*

- *$Reach_\star(\mathcal{E}) = \{\mathcal{F} \mid \exists \mathcal{E}_0, \dots, \mathcal{E}_k$ $(k \geq 0)$ s.t. $\mathcal{E}_0 = \mathcal{E}$, $\mathcal{E}_k = \mathcal{F}$, and $\mathcal{E}_i \overset{X}{\rightharpoonup}_\star \mathcal{E}_{i+1}$ $(i < k)\}$, where $\mathcal{E}_i \overset{X}{\rightharpoonup}_\star \mathcal{E}_{i+1}$ iff $\exists X \in Conf_\star(\mathcal{E}_i)\colon \mathcal{E}_{i+1} = \mathcal{E}_i \setminus X$ and $\emptyset \rightarrow_\star X$ in $\mathcal{E}_i$;*
- *$TR_\star(\mathcal{E})$ is the transition system $(Reach_\star(\mathcal{E}), \rightharpoonup_\star, \mathcal{E})$ over $\mathbb{L}_\star$, where $\mathcal{F} \overset{p}{\rightharpoonup}_\star \mathcal{F}'$ iff $\exists X \in Conf_\star(\mathcal{F})\colon \mathcal{F} \overset{X}{\rightharpoonup}_\star \mathcal{F}'$ and $p = l_\star(X)$;*
- *$TR_{whp}(\mathcal{E})$ is the transition system $(Reach_{int}(\mathcal{E}), \rightharpoonup_{whp}, \mathcal{E})$ over $\mathbb{L}_{pom}$, where $\mathcal{F} \overset{p}{\rightharpoonup}_{whp} \mathcal{F}'$ iff $\exists X, X' \in Conf_{int}(\mathcal{E})\colon \mathcal{F} = \mathcal{E} \setminus X, \mathcal{F}' = \mathcal{E} \setminus X'$, $X \rightharpoondown_{whp} X'$, and $p = l_{whp}(X')$;*
- *$TR_{hp}(\mathcal{E})$ is the transition system $(Reach_{int}(\mathcal{E}), \rightharpoonup_{hp}, \mathcal{E})$ over $\mathbb{L}_{pom}$, where $\mathcal{F} \overset{q}{\rightharpoonup}_{hp} \mathcal{F}'$ iff $\exists X, X' \in Conf_{int}(\mathcal{E})\colon \mathcal{F} = \mathcal{E} \setminus X$, $\mathcal{F}' = \mathcal{E} \setminus X'$, $[e_1 \dots e_n] \overset{q}{\rightharpoondown}_{hp} [e_1 \dots e_n e_{n+1}]$, where $[e_1 \dots e_n] \in Der(X)$, $[e_1 \dots e_n e_{n+1}] \in Der(X')$, and $q = l([e_1 \dots e_n e_{n+1}])$.*

For instance, Figures 2–4 indicate the transition systems $TR_\star(\mathcal{E})$ with the states — the residuals of the structures considered in Examples 1–3, respectively. Here, $\star = step$, if $\mathcal{E} = \mathcal{E}^{rc}$; $\star = whp$, if $\mathcal{E} = \mathcal{E}^{ep}$; and $\star = pom$, if $\mathcal{E} = \mathcal{E}^{s}$.

Theorem 1. *Given $\star \in \{int, step, pom, whp\}$, $TC_\star(\mathcal{E})$ and $TR_\star(SF(PU(\mathcal{E})))$ ($TR_\star(\mathcal{E})$) are isomorphic; however, $TC_{hp}(\mathcal{E})$ and $TR_{hp}(SF(PU(\mathcal{E})))$ ($TR_{hp}(\mathcal{E})$) are not bisimilar; where $\mathcal{E} \in \mathbb{E}_L^{rc}$ ($\mathcal{E} \in \mathbb{E}_L^{ep} \cup \mathbb{E}_L^{s}$).*

It is easy to see that even for the EP-structure $\mathcal{E}_1^{ep}$ over $L = \{a,b,c\}$, with $E_1^{ep} = L$;, $\sharp_1^{ep} = \emptyset$, $\rightarrow_1^{ep} = \{(a,c),(b,c)\}$, and the identity labeling function l_1^{ep}, $TC_{hp}(\mathcal{E}_1^{ep})$ and $TR_{hp}(\mathcal{E}_1^{ep})$ are not bisimilar.

From Lemmas 1, 2, 3, and Theorem 1 we get

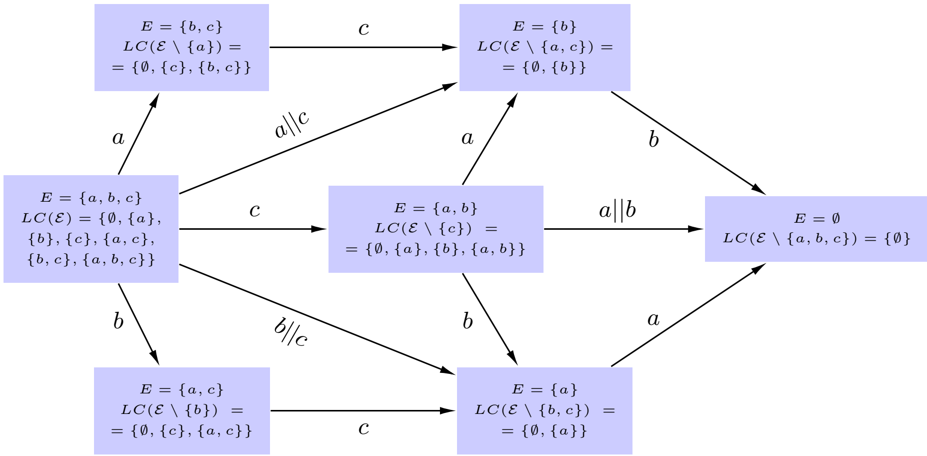

Fig. 2. The residual transition system $TR_{step}(\mathcal{E}^{rc})$

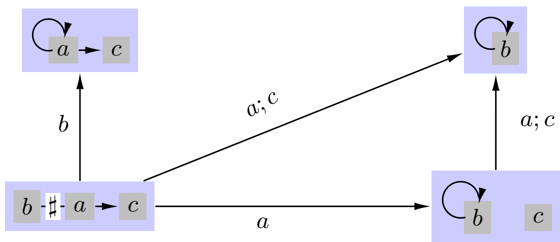

Fig. 3. The residual transition system $TR_{whp}(\mathcal{E}^{ep})$

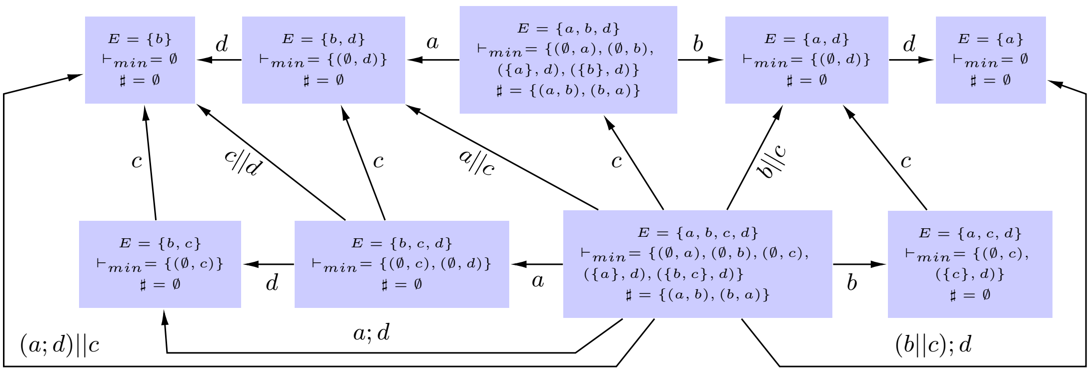

Fig. 4. The residual transition system $TR_{pom}(\mathcal{E}^s)$

Corollary 1. *Given* $\star \in \{int, step, pom, whp\}$*,*

(i) $TR_\star(\mathcal{E})$ *and* $TR_\star(SF(PU(\mathcal{RC}(\mathcal{E}))))$ *are isomorphic, if* $\mathcal{E} \in \mathbb{E}_L^{ep} \cup \mathbb{E}_L^s$*;*
(ii) $TR_\star(SF(PU(\mathcal{E})))$*,* $TR_\star(\mathcal{EP}(\mathcal{E}))$*, and* $TR_\star(\mathcal{S}(\mathcal{E}))$ *are isomorphic, if* $\mathcal{E} \in \mathbb{E}_L^{rc}$*.*

3.2 Preserving Bisimulations by the Operators $TC(\cdot)$ and $TR(\cdot)$

We first introduce bisimulation concepts on the event structure models.

Event structures $\mathcal{E}$ and $\mathcal{E}'$ from $\mathbb{E}_L$ are *interleaving, step, pomset, respectively, bisimilar* iff $TC(\mathcal{E}_\star)$ and $TC(\mathcal{E}'_\star)$ are bisimilar for $\star \in \{int, step, pom\}$, respectively. For event structures $\mathcal{E}$ and $\mathcal{E}'$ over L,

- a relation $R \subseteq Conf_{int}(\mathcal{E}) \times Conf_{int}(\mathcal{E}')$ is called *weak history preserving bisimulation* iff $(\emptyset, \emptyset) \in R$ and for any $(X, Y) \in R$ it holds:
  - there is an isomorphism between $\mathcal{E}\lceil X$ and $\mathcal{E}'\lceil Y$;
  - if $X \subseteq X'$ for some $X' \in Conf_{int}(\mathcal{E})$, then $Y \subseteq Y'$ for some $Y' \in Conf_{int}(\mathcal{E}')$ such that $(X', Y') \in R$;
  - if $Y \subseteq Y'$ for some $Y' \in Conf_{int}(\mathcal{E}')$, then $X \subseteq X'$ for some $X' \in Conf_{int}(\mathcal{E})$ such that $(X', Y') \in R$.
- a relation R consisting of triples (X, f, Y), where $X \in Conf_{int}(\mathcal{E})$, $Y \in Conf_{int}(\mathcal{E}')$, and $f : \mathcal{E}\lceil X \to \mathcal{E}'\lceil Y$ is an isomorphism, is called *history preserving bisimulation* iff $(\emptyset, \emptyset, \emptyset) \in R$ and for any $(X, f, Y) \in R$ it holds:
  - if $X \subseteq X'$ for some $X' \in Conf_{int}(\mathcal{E})$, then $Y \subseteq Y'$ for some $Y' \in Conf_{int}(\mathcal{E}')$ such that $f' \mid_X = f$ for some isomorphism $f' : X' \to Y'$, and $(X', f', Y') \in R$;
  - if $Y \subseteq Y'$ for some $Y' \in Conf_{int}(\mathcal{E}')$, then $X \subseteq X'$ for some $X' \in Conf_{int}(\mathcal{E})$ such that $f' \mid_X = f$ for some isomorphism $f' : X' \to Y'$, and $(X', f', Y') \in R$.

Theorem 2. *Given* $\mathcal{E}, \mathcal{E}' \in \mathbb{E}_L$*,* $\mathcal{E}$ *and* $\mathcal{E}'$ *are weak history preserving bisimilar iff* $TC_{whp}(\mathcal{E})$ *and* $TC_{whp}(\mathcal{E}')$ *are bisimilar;* $\mathcal{E}$ *and* $\mathcal{E}'$ *are history preserving bisimilar iff* $TC_{hp}(\mathcal{E})$ *and* $TC_{hp}(\mathcal{E}')$ *are bisimilar.*

Corollary 2. $\mathcal{E}$ *and* $\mathcal{E}'$ *are interleaving, step, pomset, weak history preserving, respectively, bisimilar iff* $TR_\star(ST(PU(\mathcal{E})))$ *and* $TR_\star(ST(PU(\mathcal{E}')))$ *(*$TR_\star(\mathcal{E})$ *and* $TR_\star(\mathcal{E}')$*) are bisimilar for* $\star \in \{int, step, pom, whp\}$*, respectively, where* $\mathcal{E}, \mathcal{E}' \in \mathbb{E}_L^{rc}$ *(*$\mathcal{E}, \mathcal{E}' \in \mathbb{E}_L^{ep} \cup \mathbb{E}_L^s$*).*

4 Concluding Remarks

In this paper, we have defined two structurally different ways of giving various (interleaving, step, pomset, weak history preserving, history preserving) transition system semantics in the context of three event-oriented models — extended prime event structures, stable event structures, and resolvable conflict structures. For each model, we have obtained an isomorphism between the corresponding transition systems for all the semantics except for history preserving one. Also, we have developed some translations of the event structures from the classes under consideration into resolvable conflict structures and back, so as to compare residual-based transition systems, constructed from the original structures, with the ones constructed from the structures obtained after translation. Further, we have demonstrated that interleaving, step, pomset, weak history preserving bisimulations are captured by the corresponding bisimulations on the transtion systems.

Work on extending our approach (e.g., to precursor [9], probabilistic [29], local [17], dynamic [1] event structures, and to labeled event structures with invisible actions) is presently under way and has yielded promising intermediate results. Another future line of research is to extend our results on comparing two kinds of transition systems to the non-pure case of resolvable conflict structures [14] and to the multiset transition relation.

References

1. Arbach, Y., Karcher, D., Peters, K., Nestmann, U.: Dynamic causality in event structures. *Lecture Notes in Computer Science* **9039** (2015) 83–97.
2. Armas-Cervantes, A., Baldan, B., Garcia-Banuelos, L.: Reduction of event structures under history preserving bisimulation. *J. Log. Algebr. Meth. Program.* 85(6) (2016) 1110-1130.
3. Baier, C., Majster-Cederbaum, M.: The connection between event structure semantics and operational semantics for TCSP. *Acta Informatica* 31 (1994).
4. Baldan, P., Corradini, A., Gadducci, F.: Domains and event structures for fusions. LICS 2017: 1-12
5. Best, E., Gribovskaya, N., Virbitskaite, I.: Configuration- and residual-based transition systems for event structures with asymmetric conflict. Proc. SofSem 2017, *Lecture Notes in Computer Science* **10877** (2018) 117-139.
6. Best, E., Gribovskaya, N., Virbitskaite, I.: From Event-Oriented Models to Transition Systems. Proc. Petri Nets 2018, *Lecture Notes in Computer Science* **10139** (2017) 132–146.
7. Boudol, G.: Flow event structures and flow nets. *Lecture Notes in Computer Science* **469** (1990) 62–95.
8. Crafa, S., Varacca, D., Yoshid, N.: Event structure semantics of parallel extrusion in the pi-calculus. *Lecture Notes in Computer Science* **7213** (2012) 225–239.
9. Fecher, H., Majster-Cederbaum, M.: Event structures for arbitrary disruption. *Fundamenta Informaticae* **68(1-2)** (2005) 103–130.
10. van Glabbeek, R.J.: History preserving process graphs. Report, Stanford University, available at http://boole.stanford.edu/ rvg/pub/abstracts/history.
11. van Glabbeek, R.J.: On the expressiveness of higher dimensional automata. *Theoretical Computer Science* **356(3)** (2006) 265–290.
12. van Glabbeek R.J., Goltz U.: Refinement of actions and equivalence notions for concurrent systems. *Acta Informatica* **37** (2001) 229–327.
13. van Glabbeek R.J., Plotkin G.D.: Configuration structures. Proc. LICS 1995: 199-209
14. van Glabbeek R.J., Plotkin G.D.: Event structures for resolvable conflict. *Lecture Notes in Computer Science* **3153** (2004) 550–561.

15. van Glabbeek R.J., Plotkin G.D.: Configuration structures, event structures and Petri nets. *Theoretical Computer Science* **410(41)** (2009) 4111-4159.
16. van Glabbeek, R.J., Vaandrager F.W.: Bundle event structures and CCSP. *Lecture Notes in Computer Science* **2761** (2003) 57–71.
17. Hoogers, P.W., Kleijn, H.C.M., Thiagarajan, P.S.: An event structure semantics for general Petri nets. *Theoretical Computer Science* **153** (1996) 129–170.
18. Katoen, J.-P.: Quantitative and qualitative extensions of event structures. PhD Thesis, Twente University, 1996.
19. Katoen, J.-P., Langerak, R., Latella, D.: Modeling systems by probabilistic process algebra: an event structures approach. *IFIP Transactions* **C-2** (1993) 253–268.
20. Keller, R.M.: Formal verification of parallel programs. Communications of the ACM, vol 19 nr 7 (1976) 371–384.
21. Langerak, R.: Bundle event structures: a non-interleaving semantics for LOTOS. Formal Description Techniques V. *IFIP Transactions* **C-10** (1993) 331–346.
22. Loogen, R., Goltz, U.: Modelling nondeterministic concurrent processes with event structures. *Fundamenta Informatica* **14(1)** (1991) 39–74.
23. Majster-Cederbaum, M., Roggenbach, M.: Transition systems from event structures revisited. *Information Processing Letters* **67(3)** (1998) 119-124.
24. Nielsen M., Plotkin G., Winskel G.: Petri nets, event structures and domains. *Theoretical Computer Science*, **13**(1) (1981) 85–08.
25. Nielsen M., Thiagarajan P.S.: Regular event structures and finite Petri nets: the conflict-free case. *Lecture Notes in Computer Science* **2360** (2002) 335–351.
26. Winskel G.: Events in computation. PhD Thesis, University of Edinburgh, 1980.
27. Winskel G.: Event structures. *Lecture Notes in Computer Science* **255** (1986) 325–392.
28. Winskel G.: Introduction to event structures. *Lecture Notes in Computer Science* **354** (1989) 364–397.
29. Winskel, G.: Distributed probabilistic and quantum strategies. *Electronic Notes in Theoretical Computer Science* **298** (2013) 403–425.