Dynamic Causality in Event Structures

Youssef Arbach, David Karcher, Kirstin Peters, Uwe Nestmann

Dynamic Causality in Event Structures*

Youssef Arbach, David Karcher, Kirstin Peters, Uwe Nestmann

Technische Universität Berlin, Germany
{youssef.arbach, david.s.karcher, kirstin.peters,
uwe.nestmann}@tu-berlin.de

Abstract Event Structures (ESs) address the representation of direct relationships between individual events, usually capturing the notions of causality and conflict. Up to now, such relationships have been static, i.e. they cannot change during a system run. Thus the common ESs only model a static view on systems. We dynamize causality such that causal dependencies between some events can be changed by occurrences of other events. We first model and study the case in which events may entail the *removal* of causal dependencies, then we consider the *addition* of causal dependencies, and finally we combine both approaches in the so-called *Dynamic Causality ESs*. For all three newly defined types of ESs, we study their expressive power in comparison to the well-known *Prime* ESs, *Dual* ESs, *Extended Bundle* ESs, and ESs for *Resolvable Conflicts*. Interestingly Dynamic Causality ESs subsume Extended Bundle ESs and Dual ESs but are incomparable with ESs for Resolvable Conflicts.

1 Introduction

Concurrency Model. Event Structures (ESs) usually address statically defined relationships that constrain the possible occurrences of events, typically represented as *causality* (for precedence) and *conflict* (for choice). An event is a single occurrence of an action; it cannot be repeated. ESs were first used to give semantics to Petri nets [14], then to process calculi [4,8], and recently to model quantum strategies and games [16]. The semantics of an ES itself is usually provided by the sets of traces compatible with the constraints, or by configuration-based sets of events, possibly in their partially-ordered variant (*posets*).

Motivation. Modern process-aware systems emphasize the need for flexibility into their design to adapt to changes in their environment [13]. One form of flexibility is the ability to change the work-flow during the runtime of the system deviating from the default path, due to changes in regulations or to exceptions. Such changes could be ad hoc or captured at the build time of the system [10]. For instance—as taken from [13]—*during the treatment process, and for a particular patient, a planned computer tomography must not be performed due to the fact that she has a cardiac pacemaker. Instead, an X-ray activity shall be performed.* In this paper, we provide a formal model that can be used for such

* Supported by the DFG Research Training Group SOAMED.

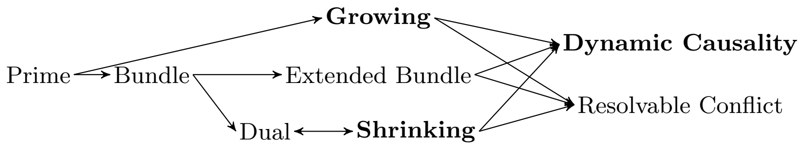

Figure 1. Landscape of Event Structures (new ESs are bold)

scenarios, showing what is the *regular* execution path and what is the *exceptional* one [10]. In the conclusion, we highlight the advantages of our model over other static-causality models w.r.t. such scenarios.

Overview. We study the idea—motivated by application scenarios—of events changing the causal dependencies of other events. In order to deal with dynamicity in causality usually duplications of events are used (see e.g. [5], where copies of the same event have the same label, but different dependencies). In this paper we want to express dynamic changes of causality more directly without duplications. We allow dependencies to change during a system run, by modifying the causality itself. In this way we avoid duplications of events, and keep the model simple and more intuitive. We separate the idea of dropping (shrinking) causality from adding (growing) causality and study each one separately first, and then combine them. In § 3 we define *Shrinking Causality Event Structures* (SESs), and compare their expressive power with other types of ESs. In § 4 we do the same for *Growing Causality Event Structures* (GESs). In § 5 we combine both concepts within the *Dynamic Causality Event Structures* (DCESs) and show that they are strictly more expressive than Extended Bundle Event Structures (EBESs) [8], which are incomparable to SESs and GESs. Although *Event Structures for Resolvable conflicts* (RCESs) [12] are shown to be more expressive than GESs and SESs, they are incomparable with DCESs. The relations among the various classes of ESs are summarised in Fig. 1, where an arrow from one class to another means that the first is less expressive than the second. In § 6 we summarize the contributions and show the limitations of other static-causality models w.r.t. our example, and conclude by future work.

Related Work. Kuske and Morin in [7] worked on local independence, using local traces. There actions can be independent from each other after a given history. Comparing to our work we provide a mechanism for independence of events, through the growing and shrinking causality, while this related work abstracts from the way actions become independent. In [12], van Glabbeek and Plotkin introduced RCESs, where conflicts can be resolved or created by the occurrence of other events. This dynamicity of conflicts is complementary to our approach. As visualized in Fig. 1 DCESs and RCESs are incomparable but—similarly to RCESs—DCESs are more expressive than many other types of ESs.

2 Technical Preliminaries

We investigate the idea of dynamically evolving dependency between events. Therefore we want to allow that the occurrence of events creates new causal dependencies between events or removes such dependencies. We base our extension on prime event structures, because they provide a very simple causality model. In the following we shortly revisit the main definitions of the types of ESs from literature we compare with. We omit the labels of events since our results are not influenced by their presence.

2.1 Prime Event Structures

A prime event structure (PES) [15] consists of a set of events and two relations describing conflicts and causal dependencies. To cover the intuition, that events causally depending on an infinite number of other events can never occur, [15] requires PESs to satisfy the *axiom of finite causes.* Additionally the enabling relation is assumed to be a partial order, i.e. is transitive and reflexive. Furthermore the concept of *conflict heredity* is required; saying that an event conflicting with another event conflicts with all its causal successors.

If we allow to add or drop causal dependencies, it is hard to maintain the conflict heredity and the transitivity and reflexivity of enabling. Because of that we do not consider the partial order property nor the axiom of conflict heredity in our definition of PESs. The same applies for the finite causes property which will be covered through finite configurations, like Def. 13 later on. Note however that the following version of PESs has the same expressive power as PESs in [15] w.r.t. to finite configurations.

Definition 1. *A* Prime Event Structure (PES) *is a triple* $\pi = (E, \#, \rightarrow)$*, where* E *is a set of* events*,* $\# \subseteq E^2$ *is an irreflexive symmetric relation (the* conflict *relation), and* $\rightarrow \subseteq E^2$ *is the* enabling *relation.*

The computation state of a process that is modeled as a PES is represented by the set of events that have occurred. Given a PES $\pi = (E, \#, \rightarrow)$ we call such sets $C \subseteq E$ that respect $\#$ and $\rightarrow$ as configurations of π.

Definition 2. *Let* $\pi = (E, \#, \rightarrow)$ *be a PES. A set of events* $C \subseteq E$ *is a* configuration *of* π *if it is* conflict-free*, i.e.* $\forall e, e' \in C \,.\, \neg\,(e \# e')$*,* downward-closed*, i.e.* $\forall e, e' \in E \,.\, e \rightarrow e' \wedge e' \in C \implies e \in C$*, and the transitive closure of the enabling relation is* acyclic*, i.e.* $\rightarrow^* \cap\, C^2$ *is free of cycles. We denote the set of configurations of* π *by* $\mathrm{C}(\pi)$*.*

An event e is called *impossible* in a PES if it does not occur in any of its configurations. Events can be impossible because of enabling cycles, or an overlapping between the enabling and the conflict relation, or because of impossible predecessors.

2.2 Bundle, Extended Bundle and Dual Event Structures

PESs are simple but also limited. They do not allow to describe optional or conditional enabling of events. Bundle event structures (BESs)—among others—were designed to overcome these limitations [8]. Enabling of events is based on

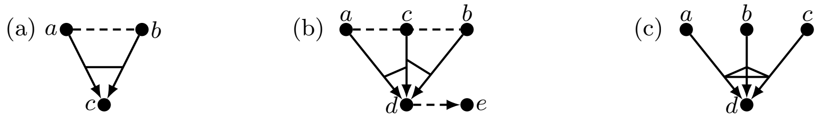

Figure 2. A Bundle ES, an Extended Bundle ES, and a Dual ES.

bundles which are pairs (X, e), denoted as $X \mapsto e$, where X is a set of events and e is the event pointed by that bundle. A bundle is satisfied when one event of X occurs. An event is enabled when all bundles pointing to it are satisfied. This disjunctive causality allows for optionality in enabling events.

Definition 3. *A* Bundle Event Structure (BES) *is a triple* $\beta = (E, \#, \mapsto)$, *where* E *is a set of* events, $\# \subseteq E^2$ *is an irreflexive symmetric relation (the* conflict *relation), and* $\mapsto \subseteq \mathcal{P}(E) \times E$ *is the* enabling *relation, such that for all* $X \subseteq E$ *and* $e \in E$ *the bundle* $X \mapsto e$ *implies that for all* $e_1, e_2 \in X$ *with* $e_1 \neq e_2$ *it holds* $e_1 \# e_2$ *(Stability).*

Figure 2 (a) shows an example of a BES. The solid arrows denote causality, i.e. reflect the enabling relation, where the bar between the arrows indicates a bundle, and the dashed line denotes a mutual conflict.

A configuration of a BES is again a conflict-free set of events that is downward-closed. Therefore the stability condition avoids causal ambiguity [9]. To exclude sets of events that result from enabling cycles we use traces. For a sequence $t = e_1 \cdots e_n$ of events let $\bar{t} = \{e_1, \ldots, e_n\}$ and $t_i = e_1 \cdots e_i$. Let ϵ denote the empty sequence.

Definition 4. *Let* $\beta = (E, \#, \mapsto)$ *be a BES. A* trace *is a sequence of distinct events* $t = e_1 \cdots e_n$ *with* $\bar{t} \subseteq E$ *such that* $\forall 1 \leq i, j \leq n \,.\, \neg (e_i \# e_j)$ *and such that* $\forall 1 \leq i \leq n \,.\, \forall X \subseteq E \,.\, X \mapsto e_i \implies \overline{t_{i-1}} \cap X \neq \emptyset$.

A set of events $C \subseteq E$ is a *configuration* of β if there is a trace t such that $C = \bar{t}$. This trace-based definition of a configuration will be the same for Extended Bundle and Dual ESs. Let $\mathrm{T}(\beta)$ denote the set of traces and $\mathrm{C}(\beta)$ the set of configurations of β.

Partially ordered sets, abbreviated as posets, are used as a semantic model for different kinds of ESs and other concurrency models (see e.g. [11]). In contrast to configurations, a poset does not only record the set of events that happened, but also captures the precedence relations between the events. Formally a poset is a pair $(A, \leq)$, where A is a finite set of *events* and $\leq$ is a *partial order* over A.

A poset represents a set of system runs, differing for permutation of independent events. To describe the semantics of the entire ES, families of posets [11] with a prefix relation are used. According to Rensink in [11], families of posets form a convenient underlying model for models of concurrency, and are more expressive than families of configurations.

To obtain the posets of a BES, wndow each of its configurations with a partial order. Let $\beta = (E, \#, \mapsto)$ be a BES and $C \in \mathrm{C}(\beta)$, and $e, e' \in C$. Then $e \prec_C e'$

if $\exists X \subseteq E \,.\, e \in X \wedge X \mapsto e'$. Let $\leq_C$ be the reflexive and transitive closure of $\prec_C$. It is proved in [8] that $\leq_C$ is a partial order over C. Let $\mathrm{P}(\beta)$ denote the set of posets of β.

Let x and y be two ESs of arbitrary kind on which posets are defined. We denote that x and y have the same set of posets by $x \simeq_{\mathrm{p}} y$. Note that for BESs, EBES, and DESs families of posets are the most discriminating semantics studied in the literature. So, in these cases, we consider two ESs as behaviorally equivalent if they have the same set of posets.

The first extension of BESs we consider are *Extended Bundle Event Structures* (EBESs) from [8]. The conflict relation # is replaced by a *disabling* relation. An event e_1 disables another event e_2, means once e_1 occurs e_2 cannot occur anymore. The symmetric conflict # can be modeled through mutual disabling. Therefore EBESs are a generalization of BESs, and thus are more expressive [8].

Definition 5. *An* Extended Bundle Event Structure (EBES) *is a triple* $\xi = (E, \rightsquigarrow, \mapsto)$*, where* E *is a set of* events*,* $\rightsquigarrow \subseteq E^2$ *is the irreflexive* disabling *relation, and* $\mapsto \subseteq \mathcal{P}(E) \times E$ *is the* enabling *relation, such that for all* $X \subseteq E$ *and* $e \in E$ *the bundle* $X \mapsto e$ *implies that for all* $e_1, e_2 \in X$ *with* $e_1 \neq e_2$ *it holds* $e_1 \rightsquigarrow e_2$ *(Stability).*

Stability ensures that two distinct events of a bundle set are in mutual disabling. Figure 2 (b) shows an EBES with the two bundles $\{a, c\} \mapsto d$ and $\{b, c\} \mapsto d$. The dashed lines denote again mutual disabling as required by stability. A disabling $d \rightsquigarrow e$, to be read 'e disables d', is represented by a dashed arrow.

Definition 6. *Let* $\xi = (E, \rightsquigarrow, \mapsto)$ *be an EBES. A* trace *is a sequence of distinct events* $t = e_1 \cdots e_n$ *with* $\overline{t} \subseteq E$ *such that* $\forall 1 \leq i, j \leq n \,.\, e_i \rightsquigarrow e_j \implies i < j$ *and* $\forall 1 \leq i \leq n \,.\, \forall X \subseteq E \,.\, X \mapsto e_i \implies \overline{t_{i-1}} \cap X \neq \emptyset$*.*

We adapt the definitions of configurations and traces of BESs accordingly. For $C \in \mathrm{C}(\xi)$ and $e, e' \in C$, let $e \prec_C e'$ if $\exists X \subseteq E \,.\, e \in C \wedge X \mapsto e'$ or if $e \rightsquigarrow e'$. Again $\leq_C$ denotes the reflexive and transitive closure of $\prec_C$, and $\mathrm{P}(\xi)$ denotes the set of posets of ξ.

Dual Event Structures (DESs) are the second extension of BES examined here. They are obtained by dropping the stability condition. This leads to causal ambiguity, i.e. given a trace and one of its events, it is not always possible to determine what caused this event. The definition of DESs varies between [6] (based on EBESs) and [9] (based on BESs). Here we rely on the version of [9].

Definition 7. *A* Dual Event Structure (DES) *is a triple* $\delta = (E, \#, \mapsto)$*, where* E *is a set of* events*,* $\# \subseteq E^2$ *is an irreflexive symmetric relation (the* conflict *relation), and* $\mapsto \subseteq \mathcal{P}(E) \times E$ *is the* enabling *relation.*

Figure 2 (c) shows a DES with one bundle, namely $\{a, b, c\} \mapsto d$, and without conflicts. Again the definitions of configurations and traces are exactly the same as in BESs (cf. Def. 4), therefore we omit them here.

Because of the causal ambiguity, the definition of $\leq_C$ is difficult and the behavior of a DES w.r.t. a configuration cannot be described by a single poset anymore. [9] illustrates that there are different possible interpretations of causality. The authors defined five different intentional posets: liberal, bundle satisfaction, minimal, early and late posets. They show the equivalence of the behavioral semantics, and that the early causality and trace equivalence coincide. Thus we concentrate on early causality. The remaining intentional partial order semantics are discussed in [1]. To capture causal ambiguity we have to consider all traces of a configuration to obtain its posets. Below U_1 is earlier than U_2 if the largest index in $U_1 \setminus U_2$ is smaller than the largest index in $U_2 \setminus U_1$ [9].

Definition 8. *Let $\delta = (E, \#, \mapsto)$ be a DES, $t = e_1 \cdots e_n$ one of its traces, $1 \leq i \leq n$, and $X_1 \mapsto e_i, \ldots, X_m \mapsto e_i$ all bundles pointing to e_i. A set U is a cause of e_i in t if $\forall e \in U \,.\, \exists 1 \leq j < i \,.\, e = e_j$, $\forall 1 \leq k \leq m \,.\, X_k \cap U \neq \emptyset$, and U is the earliest set satisfying the previous two conditions. Let $\mathrm{P_d}(t)$ be the set of posets obtained this way for t.*

2.3 Event Structures for Resolvable Conflicts

Event Structures for Resolvable Conflicts (RCES) were introduced in [12] to generalize former types of ESs and to give semantics to general Petri Nets. They allow to model the case where a and c cannot occur together until b takes place, i.e. initially a and c are in a conflict until the occurrence of b resolves this conflict. An RCES consists of a set of events and an enabling relation between sets of events. Here the enabling relation also models conflicts between events. The behavior is defined by a transition relation between sets of events that is derived from the enabling relation $\vdash$.

Definition 9. *An* Event Structure for Resolvable Conflicts (RCES) *is a pair $\rho = (E, \vdash)$, where E is a set of* events *and $\vdash \subseteq \mathcal{P}(E)^2$ is the* enabling relation*.*

In [12] several versions of configurations are defined. Here we consider only configurations which are both reachable and finite.

Definition 10. *Let $\rho = (E, \vdash)$ be an RCES and $X, Y \subseteq E$. Then $X \rightarrow_{\mathrm{rc}} Y$ if $(X \subseteq Y \wedge \forall Z \subseteq Y \,.\, \exists W \subseteq X \,.\, W \vdash Z)$. The set of* configurations *of ρ is defined as $\mathrm{C}(\rho) = \{ X \subseteq E \mid \emptyset \rightarrow_{\mathrm{rc}}^{*} X \wedge X \text{ is finite} \}$, where $\rightarrow_{\mathrm{rc}}^{*}$ is the reflexive and transitive closure of $\rightarrow_{\mathrm{rc}}$.*

As an example consider the RCES $\rho = (E, \vdash)$, where $E = \{a, b, c\}$, $\{b\} \vdash \{a, c\}$, and $\emptyset \vdash X$ iff $X \subseteq E$ and $X \neq \{a, c\}$. It models the above described initial conflict between a and c that can be resolved by b. In Fig. 3 (ρ) the respective transition graph is shown, i.e. the nodes are all reachable configurations of ρ and the directed edges represent $\rightarrow_{\mathrm{rc}}$. Note, because of $\{a, c\} \subset \{a, b, c\}$ and $\emptyset \not\vdash \{a, c\}$, there is no transition from $\emptyset$ to $\{a, b, c\}$.

We consider two RCESs as equivalent if they have the same transition graphs. Note that, since we consider only reachable configurations, the transition equivalence defined below is denoted as reachable transition equivalence in [12].

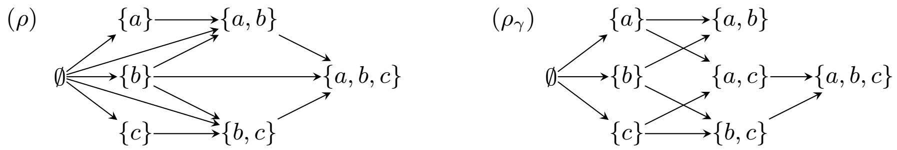

Figure 3. Transition graphs of RCESs with resolvable conflict (ρ) and disabling (ρ_γ).

Definition 11. *Two RCESs* $\rho = (E, \rightarrow_{\mathrm{rc}})$ *and* $\rho' = (E', \rightarrow'_{\mathrm{rc}})$ *are* transition equivalent*, denoted by* $\rho \simeq_{\mathrm{t}} \rho'$*, if* $E = E'$ *and* $\rightarrow_{\mathrm{rc}} \cap (\mathrm{C}(\rho))^2 = \rightarrow'_{\mathrm{rc}} \cap (\mathrm{C}(\rho'))^2$.

Again we adapt the notion of transition equivalence to arbitrary types of ESs with a transition relation. Let x and y be two arbitrary types of ESs on that a transition relation is defined. We denote the fact that x and y have the same transition graphs by $x \simeq_{\mathrm{t}} y$. Note that for RCESs, transition equivalence is the most discriminating semantics studied in the literature. So we consider two RCESs as behavioral equivalent if they have the same transition graphs.

3 Shrinking Causality

Now we add a new relation which represents the removal of causal dependencies as a ternary relation between events $\rhd \subseteq E^3$. For instance $(a, c, b) \in \rhd$, denoted as $[a \rightarrow b] \rhd c$, models that a is dropped from the set of causal predecessors of b by the occurrence of c. The dropping is visualized in Fig. 4(a) by a dashed arrow with empty head from the initial cause $a \rightarrow b$ to its dropper c. We add this relation to PESs and denote the result as shrinking causality event structures.

Definition 12. *A* Shrinking Causality Event Structure (SES) *is a pair* $\sigma = (\pi, \rhd)$*, where* $\pi = (E, \#, \rightarrow)$ *is a PES and* $\rhd \subseteq E^3$ *is the* shrinking causality *relation such that* $[e \rightarrow e''] \rhd e'$ *implies* $e \rightarrow e''$ *for all* $e, e', e'' \in E$.

Sometimes we expand $(\pi, \rhd)$ and write $(E, \#, \rightarrow, \rhd)$. For $[c \rightarrow t] \rhd m$ we call m the modifier, t the target, and c the contribution. We denote the set of all modifiers dropping $c \rightarrow t$ by $[c \rightarrow t] \rhd$. We refer to the set of dropped causes of an event w.r.t. a specific history by the function $\mathrm{dc} : \mathcal{P}(E) \times E \rightarrow \mathcal{P}(E)$ defined as: $\mathrm{dc}(H, e) = \{e' \mid \exists d \in H\ .\ [e' \rightarrow e] \rhd d\}$. We refer to the initial causes of an event by the function $\mathrm{ic} : E \rightarrow \mathcal{P}(E)$ such that: $\mathrm{ic}(e) = \{e' \mid e' \rightarrow e\}$. The semantics of a SES can be defined based on posets similar to BESs, EBESs, and DESs, or based on a transition relation similar to RCESs. We consider both.

Definition 13. *Let* $\sigma = (E, \#, \rightarrow, \rhd)$ *be a SES.*

- *A* trace *of* σ *is a sequence of distinct events* $t = e_1 \cdots e_n$ *with* $\bar{t} \subseteq E$ *such that* $\forall 1 \leq i, j \leq n\ .\ \neg(e_i \# e_j)$ *and* $\forall 1 \leq i \leq n\ .\ \big(\mathrm{ic}(e_i) \setminus \mathrm{dc}(\overline{t_{i-1}}, e_i)\big) \subseteq \overline{t_{i-1}}$. $C \subseteq E$ *is a* traced-based configuration *of* σ *if there is* t *such that* $C = \bar{t}$. *Let* $\mathrm{C}_{Tr}(\sigma)$ *be the set of traced-based configurations,* $\mathrm{T}(\sigma)$ *the set of traces of* σ.

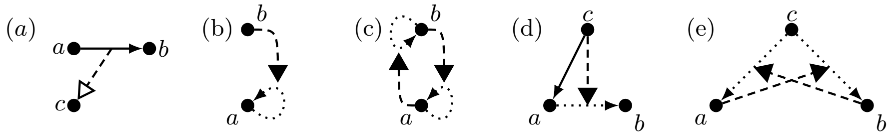

Figure 4. A SES and GESs modeling disabling, conflict, temporary disabling, and resolvable conflicts.

- *Let* $t = e_1 \cdots e_n \in \mathrm{T}(\sigma)$ *and* $1 \leq i \leq n$*. A set* U *is a cause of* e_i *in* t *if* $\forall e \in U \,.\, \exists 1 \leq j < i \,.\, e = e_j$, $(\mathrm{ic}(e_i) \setminus \mathrm{dc}(U, e_i)) \subseteq U$*, and* U *is the earliest set satisfying the previous two conditions. Let* $\mathrm{P_s}(t)$ *be the set of posets obtained this way for* t*.*
- *Let* $X, Y \subseteq E$*. Then* $X \rightarrow_{\mathrm{s}} Y$ *if* $X \subseteq Y$, $\forall e, e' \in Y \,.\, \neg(e \# e')$*, and* $\forall e \in Y \setminus X \,.\, (\mathrm{ic}(e) \setminus \mathrm{dc}(X, e)) \subseteq X$*.*
- *The set of all configurations of* σ *is* $\mathrm{C}(\sigma) = \{X \subseteq E \mid \emptyset \rightarrow_{\mathrm{s}}^{*} X \wedge X \text{ is finite } \}$*, where* $\rightarrow_{\mathrm{s}}^{*}$ *is the reflexive and transitive closure of* $\rightarrow_{\mathrm{s}}$*.*

The combination of initial and dropped causes ensures that for each $e_i \in \bar{t}$, all its initial causes are either preceding e_i or dropped by other events preceding e_i. Note that as for DESs we concentrate on early causality. We consider the reachable and finite configurations w.r.t. to $\rightarrow_{\mathrm{s}}$ as well as configurations based on the traces. Note that both definitions coincide.

To show that $\simeq_{\mathrm{p}}$ and $\simeq_{\mathrm{t}}$ coincide on SESs we make use of the result that $\simeq_{\mathrm{p}}$ and trace equivalence coincide on DES (compare to [9]) and show that SESs are as expressive as DESs. Consider the shrinking-causality $[c \rightarrow t] \triangleright d$. It models the case that initially t causally depends on c which can be dropped by the occurrence of d. Thus for t to be enabled either c occurs or d does. This is a disjunctive causality as modeled by DESs. In fact $[c \rightarrow t] \triangleright d$ corresponds to the bundle $\{c, d\} \mapsto t$. We prove that we can map each SES into a DES with the same behavior and vice versa. To translate a SES into a DES we create a bundle for each initial causal dependence and add all its droppers to the bundle set.

In the opposite direction we map each DES into a set of similar SESs such that each SES in this set has the same behavior as the DES. Intuitively we have to choose an initial dependency for each bundle from its set, and to translate the rest of the bundle set into droppers for that dependency. Unfortunately the bundles that point to the same event are not necessarily disjoint. Consider for example $\{a, b\} \mapsto e$ and $\{b, c\} \mapsto e$. If we choose $b \rightarrow e$ as initial dependency for both bundles to be dropped as $[b \rightarrow e] \triangleright a$ and $[b \rightarrow e] \triangleright c$, then $\{a, e\}$ is a configuration of the resulting SES but not of the original DES. So we have to ensure that we choose distinct events as initial causes for all bundles pointing to the same event. Thus for each bundle we choose a fresh event as initial cause, make it impossible by a self-loop, and add all events of the bundle as droppers. Note that to translate a DES into a SES we have to introduce additional events, i.e. it is not always possible to translate a DES into a SES without additional impossible events. All proofs can be found in [1].

Theorem 1. *SESs are as expressive as DESs.*

In [1] we show that each SES and its translation as well as each DES and its translation have the same set of posets considering not only early but also liberal, minimal, and late causality. Thus the concepts of SESs and DESs are not only behaviorally equivalent but—except for the additional impossible events—also structurally closely related.

Note that $\simeq_\mathrm{p}$, $\simeq_\mathrm{t}$, and trace-equivalence coincide on SES.

Theorem 2. *Let σ, σ' be two SESs. Then $\sigma \simeq_\mathrm{p} \sigma'$ iff $\sigma \simeq_\mathrm{t} \sigma'$ iff $\mathrm{T}(\sigma) = \mathrm{T}(\sigma')$.*

As shown above SESs allow to model disjunctive causality. As an example consider the dropping of a causality as in Fig. 4(a). Such a disjunctive causality is not possible in EBESs. On the other hand the asymmetric conflict of an EBES cannot be modeled with a SES. As an example consider Fig. 2 (b),where e cannot precede d.

Theorem 3. *SESs and EBESs are incomparable.*

SESs are strictly less expressive than RCESs, because each SES can be translated into a transition-equivalent RCES, and on the other hand there are RCESs that cannot be translated into a transition-equivalent SES. As a counterexample we use the RCES $\rho_\sigma = (\{e, f\}, \{\emptyset \vdash \emptyset, \emptyset \vdash \{e\}, \emptyset \vdash \{f\}, \{e\} \vdash \{e, f\}\})$ that captures disabling in an EBES.

Theorem 4. *SESs are strictly less expressive than RCESs.*

4 Growing Causality

As in SESs we base our extension for growing causality on PESs. We add the new relation $\blacktriangleright \subseteq E^3$, where $(a, c, b) \in \blacktriangleright$, denoted as $c \blacktriangleright [a \to b]$, models that c adds a as a cause for b. Thus c is a condition for the causal dependency $a \to b$.

The adding is visualized in Fig. 4(d) by a dashed line with a filled head from the modifier c to the added dependency $a \to b$, which is dotted denoting that this dependency does not exist initially (In this example there is an additional causality $c \to a$).

Definition 14. *A* Growing Causality Event Structure (GES) *is a pair $\gamma = (\pi, \blacktriangleright)$, where $\pi = (E, \#, \to)$ is a PES and $\blacktriangleright \subseteq E^3$ is the* growing causality *relation such that $\forall e, e', e'' \in E \,.\, e' \blacktriangleright [e \to e''] \implies \neg(e \to e'')$.*

We refer to the causes added to an event w.r.t. a specific history by the function $\mathrm{ac} : \mathcal{P}(E) \times E \to \mathcal{P}(E)$, defined as $\mathrm{ac}(H, e) = \{e' \mid \exists a \in H \,.\, a \blacktriangleright [e' \to e]\}$, and to the initial causality by the function ic as defined in § 3. Similar to the RCESs the behavior of a GES can be defined by a transition relation. Thus we consider two GESs as equally expressive if they are transition-equivalent.

Definition 15. *Let $\gamma = (E, \#, \rightarrow, \blacktriangleright)$ be a GES.*

- *A* trace *of γ is a sequence of distinct events $t = e_1 \cdots e_n$ with $\bar{t} \subseteq E$ such that $\forall 1 \leq i, j \leq n \,.\, \neg(e_i \# e_j)$ and $\left(\mathrm{ic}(e_i) \cup \mathrm{ac}\left(\overline{t_{i-1}}, e_i\right)\right) \subseteq \overline{t_{i-1}}$ for all $i \leq n$. Then $C \subseteq E$ is a* trace-based configuration *of γ if there is a trace t such that $C = \bar{t}$. The set of traces of γ is denoted by $\mathrm{T}(\gamma)$ and the set of its trace-based configurations is denoted by $\mathrm{C}_{Tr}(\gamma)$.*
- *Let $X, Y \subseteq E$. Then $X \rightarrow_{\mathrm{g}} Y$ if $X \subseteq Y$, $\forall e, e' \in Y \,.\, \neg(e \# e')$, $\forall e \in Y \setminus X \,.\, (\mathrm{ic}(e) \cup \mathrm{ac}(X, e)) \subseteq X$, and $\forall t, m \in Y \setminus X \,.\, \forall c \in E \,.\, m \blacktriangleright [c \rightarrow t] \implies (c \in X \vee m \in \{c, t\})$.*
- *The set of all configurations of γ is $\mathrm{C}(\gamma) = \left\{ X \subseteq E \mid \emptyset \rightarrow_{\mathrm{g}}^{*} X \wedge X \text{ is finite} \right\}$, where $\rightarrow_{\mathrm{g}}^{*}$ is the reflexive and transitive closure of $\rightarrow_{\mathrm{g}}$.*

The last condition in the transition definition prevents the concurrent occurrence of a target and its modifier since they are not independent. One exception is when the contribution has already occurred; in that case, the modifier does not change the target's predecessors. It also captures the trivial case of self adding, i.e. when a target adds a contribution to itself or a modifier adds itself to a target. Again we consider the reachable and finite configurations, and show in [1] that the definitions of reachable and trace-based configurations coincide.

Disabling as defined in EBESs or the asymmetric event structure of [3] can be modeled by $\blacktriangleright$. For example $b \rightsquigarrow a$ can be modeled by $b \blacktriangleright [a \rightarrow a]$ as depicted in Fig. 4 (b). Conflicts can be modeled by $\blacktriangleright$ through mutual disabling, as depicted in Fig. 4 (c), and thus the conflict relation can be omitted in this ES model.

In inhibitor event structures [2] there is a kind of disabling, where an event e can be disabled by another event d until an event out of a set X occurs. This kind of temporary disabling provides disjunction in the re-enabling that cannot be modeled in GESs but in DCESs (cf. the next section). However temporary disabling without a disjunctive re-enabling can be modeled by a GES as in Fig. 4 (d).

Also resolvable conflicts can be modeled by a GES. For example the GES in Fig. 4 (e) with $a \blacktriangleright [c \rightarrow b]$ and $b \blacktriangleright [c \rightarrow a]$ models a conflict between a and b that can be resolved by c. Note that this example depends on the idea that a modifier and its target cannot occur concurrently (cf. Def. 15). Note also that resolvable conflicts are a reason why families of configurations cannot be used to define the semantics of GESs or RCESs.

As shown in Fig. 4 (b) GESs can model disabling. Nevertheless EBESs and GESs are incomparable, because GESs cannot model the disjunction in the enabling relation that EBESs inherit from BESs. On the other hand EBESs cannot model conditional causal dependencies. Thus GESs are incomparable to BESs as well as EBESs.

Theorem 5. *GESs are incomparable to BESs and EBESs.*

GESs are also incomparable to SESs, because the adding of causes cannot be modeled by SESs. As a counterexample we use the GES of Fig. 4 (c). Then since BESs are incomparable to GESs, BESs are less expressive than DESs, and DESs are as expressive as SESs, we conclude that GESs and SESs are incomparable.

Theorem 6. *GESs and SESs are incomparable.*

As illustrated in Fig. 4 (d) GESs can model resolvable conflicts. Nevertheless they are strictly less expressive than RCESs, because each GES can be translated into a transition equivalent RCES and on the other hand there exists no transition equivalent GES for the RCES $\rho_\gamma = (\{a, b, c\}, \vdash)$ that is given by the second transition graph in Fig. 3. It models the case, where after a and b the event c becomes impossible, i.e. it models disabling by a set instead of a single event.

Theorem 7. *GESs are strictly less expressive than RCESs.*

5 Dynamic Causality

Up to now we have investigated shrinking, and growing causality separately. In this section we combine them and examine the resulting expressiveness.

Definition 16. *A* Dynamic Causality Event Structure (DCES) *is a triple* $\Delta = (\pi, \triangleright, \blacktriangleright)$*—expanded* $(E, \#, \rightarrow, \triangleright, \blacktriangleright)$*—, where* $\pi = (E, \#, \rightarrow)$ *is a PES,* $\triangleright \subseteq E^3$ *is the shrinking causality relation, and* $\blacktriangleright \subseteq E^3$ *is the growing causality relation such that for all* $e, e', e'' \in E$*: 1.* $[e \rightarrow e''] \triangleright e' \wedge \nexists m \in E \,.\, m \blacktriangleright [e \rightarrow e''] \Longrightarrow e \rightarrow e''$ *2.* $e' \blacktriangleright [e \rightarrow e''] \wedge \nexists m \in E \,.\, [e \rightarrow e''] \triangleright m \Longrightarrow \neg(e \rightarrow e'')$ *3.* $e' \blacktriangleright [e \rightarrow e''] \Longrightarrow \neg([e \rightarrow e''] \triangleright e')$*.*

Conditions 1 and 2 are just a generalization of the conditions in Defs. 12 and 14 respectively. If there are droppers and adders for the same causal dependency we do not specify whether this dependency is contained in $\rightarrow$, because the semantics depends on the order in which the droppers and adders occur. Condition 3 prevents that a modifier adds and drops the same cause for the same target.

The order of occurrence of droppers and adders determines the causes of an event. For example assume $a \blacktriangleright [c \rightarrow t]$ and $[c \rightarrow t] \triangleright d$, then after ad, t does not depend on c, whereas after da, t depends on c. Thus configurations like $\{a, d\}$ are not expressive enough to represent the state of such a system (cf. Lem. 1).

Therefore in a DCES a state is a pair of a configuration C and a causal state function cs, which computes the causal predecessors of an event, that are still needed.

Definition 17. *Let* $\Delta = (E, \#, \rightarrow, \triangleright, \blacktriangleright)$ *be a DCES. The function* $\mathrm{mc} : \mathcal{P}(E) \times E \rightarrow \mathcal{P}(E)$ *denotes the* maximal causality *that an event can have after some history* $C \subseteq E$*, and is defined as* $\mathrm{mc}(C, e) = \{e' \in E \setminus C \mid e' \rightarrow e \vee \exists a \in C \,.\, a \blacktriangleright [e' \rightarrow e]\}$*. A* state *of* Δ *is a pair* (C, cs) *where* $\mathrm{cs} : E \setminus C \rightarrow \mathcal{P}(E \setminus C)$ *such that* $C \subseteq E$ *and* $\mathrm{cs}(e) \subseteq \mathrm{mc}(C, e)$*. We denote* cs *as* causality state function*, which shows for an event e that did not occur, which events are still missing such that e is enabled. An initial state of* Δ *is* $S_0 = (\emptyset, \mathrm{cs_i})$*, where* $\mathrm{cs_i}(e) = \{e' \in E \mid e' \rightarrow e\}$*.*

Note that S_0 is the only state with an empty set of events; for other sets of events there can be multiple states. The behavior of a DCES is defined by the transition relation on its reachable states with finite configurations.

Definition 18. *Let $\Delta = (E, \#, \rightarrow, \triangleright, \blacktriangleright)$ be a DCES and $C, C' \subseteq E$. Then $(C, \mathrm{cs}) \rightarrow_{\mathrm{d}} (C', \mathrm{cs}')$ if:*
1. $C \subseteq C'$ *2.* $\forall e, e' \in C' \,.\, \neg(e \# e')$ *3.* $\forall e \in C' \setminus C \,.\, \mathrm{cs}(e) = \emptyset$
4. $\forall e, e' \in E \setminus C' \,.\, e' \in \mathrm{cs}(e) \setminus \mathrm{cs}'(e) \Longrightarrow [e' \rightarrow e] \triangleright \cap (C' \setminus C) \neq \emptyset$
5. $\forall e, e' \in E \setminus C' \,.\, [e' \rightarrow e] \triangleright \cap (C' \setminus C) \neq \emptyset \Longrightarrow e' \notin \mathrm{cs}'(e)$
6. $\forall e \in E \setminus C' \,.\, e' \in \mathrm{cs}'(e) \setminus \mathrm{cs}(e) \Longrightarrow \blacktriangleright [e' \rightarrow e] \cap (C' \setminus C) \neq \emptyset$
7. $\forall e, e' \in E \setminus C' \,.\, \blacktriangleright [e' \rightarrow e] \cap (C' \setminus C) \neq \emptyset \Longrightarrow e' \in \mathrm{cs}'(e)$
8. $\forall e, e' \in E \setminus C \,.\, [e' \rightarrow e] \triangleright \cap (C' \setminus C) = \emptyset \vee \blacktriangleright [e' \rightarrow e] \cap (C' \setminus C) = \emptyset$
9. $\forall t, m \in C' \setminus C \,.\, \forall c \in E \,.\, m \blacktriangleright [c \rightarrow t] \implies (c \in C \vee m \in \{c, t\})$.

Condition 1 ensures the accumulation of events. Condition 2 ensures conflict freeness. Condition 3 ensures that only events which are enabled after C can take place in C'. Condition 4 ensures that, if a cause disappears, there has to be a dropper of it. The same is ensured by Condition 6 for appearing causes. Condition 5 ensures that if there are adders, the cause has to appear in the new causal state, unless it occurred. Similarly, Condition 7 ensures, that causes disappear, when there are droppers. To keep the theory simple, Condition 8 avoids race conditions; it forbids the occurrence of an adder and a dropper of the same causal dependency within one transition. Condition 9 ensures that DCESs coincide with GESs.

Definition 19. *Let Δ be a DCES. The set of (reachable) states of Δ is defined as $\mathrm{S}(\Delta) = \{(X, \mathrm{cs}_X) \mid S_0 \rightarrow_{\mathrm{d}}^{*} (X, \mathrm{cs}_X) \wedge X$ is finite $\}$, where $\rightarrow_{\mathrm{d}}^{*}$ is the reflexive and transitive closure of $\rightarrow_{\mathrm{d}}$.*

Two DCESs $\Delta = (E, \#, \rightarrow, \triangleright, \blacktriangleright)$ and $\Delta' = (E', \#', \rightarrow', \triangleright', \blacktriangleright')$ are state transition equivalent*, denoted by $\Delta \simeq_{\mathrm{s}} \Delta'$, if $E = E'$ and $\rightarrow_{\mathrm{d}} \cap (\mathrm{S}(\Delta))^2 = \rightarrow'_{\mathrm{d}} \cap (\mathrm{S}(\Delta'))^2$.*

Lemma 1. *There are DCESs that are transition equivalent but not state transition equivalent.*

Because of the previous Lemma, we consider the more discriminating equivalence $\simeq_{\mathrm{s}}$—instead of $\simeq_{\mathrm{t}}$—to compare DCESs and to compare with DCESs. To compare with RCESs, we use the counterexample ρ_γ of Fig. 3 to show that not for every RCES there is a transition-equivalent DCES. Moreover RCESs cannot distinguish between different causality states of one configuration (cf. Lem. 1). Consequently DCESs and RCESs are incomparable.

Theorem 8. *DCESs and RCESs are incomparable.*

By construction, DCESs are at least as expressive as GESs and SESs. To embed a SES (or GES) into a DCES it suffice to choose $\blacktriangleright = \emptyset$ (or $\triangleright = \emptyset$). Furthermore are DCESs incomparable to RCESs which are strictly more expressive than GESs and SESs. Thus DCESs are strictly more expressive then GESs and SESs.

Theorem 9. *DCESs are strictly more expressive than GESs and SESs.*

To compare with EBESs, we use the disabling of GESs, and the disjunctive causality of SESs. The translation of an EBES into a DCES is formally defined in

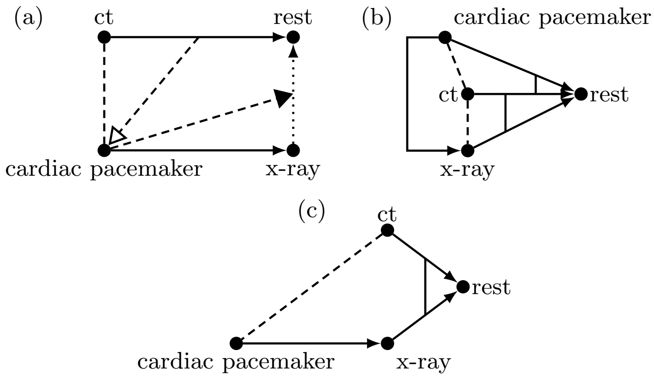

Figure 5. A DCES, a BES, and a DES modeling the medical example.

[1], where disabling uses self-loops of target events, while droppers use auxiliary impossible events, not to intervene with the disabling. Besides we construct posets for the configurations of the translation, and compare them with those of the original EBES. In this way we prove that DCESs are at least as expressive as EBESs. But since EBESs cannot model the disjunctive causality—without a conflict—of SESs which are included in DCESs, the following result holds.

Theorem 10. *DCESs are strictly more expressive than EBESs.*

6 Conclusions

We study the idea that causality may change during system runs in event structures. For this, we enhance a simple type of ESs—the PES—by means of additional relations capturing the changes in events' dependencies, driven by the occurrence of other events.

First, in § 3, we limit our concern to the case where dependencies can only be dropped. We call the new resulting event structure Shrinking Causality ES (SES). In that section, we show that the exhibited dynamic causality can be expressed through a completely static perspective, by proving equivalence between SESs and DESs. By such a proof, we do not only show the expressive power of our new ES, but also the big enhancement in expressive power (w.r.t. PESs) gained by adding only this one relation.

Later on, in § 4, we study the complementary style where dependencies can be added to events, resulting in Growing Causality ES (GES). We show that the growing causality can model both permanent and temporary disabling. Besides, it can be used to resolve conflicts and, furthermore, to force conflicts. Unlike the SESs, the GESs are not directly comparable to other types of ESs from the literature, except for PESs; one reason is that they provide a conjunctive style of causality, another is their ability to express conditioning in causality.

Finally, in § 5, we combine both approaches of dynamicity with a new type of event structures, which we called the Dynamic Causality ES (DCES). Therein a dependency can be both added and dropped. For this new type of ESs the following two—possibly surprising—facts can be observed: (1) There are types of ESs that are incomparable to both SESs and GESs, but that are comparable to (here: strictly less expressive than) DCESs, i.e. the combination of SESs and GESs; one such type is EBESs. (2) Though SESs and GESs are strictly less expressive than RCESs, their combination—the newly defined DCESs—is incomparable to RCESs, or any other type of ESs with a static causality.

To highlight the pragmatic advantages of dynamic-causality ESs over their equivalent and non-equivalent competitor ESs, we go back to our example mentioned in the motivation. Reichert et al. in [10] emphasize that the model of such processes should distinguish between the regular execution path and the exceptional one. Accordingly, they define two labels, *REGULAR* and *EXCEPTIONAL*, to be assigned to tasks. Fig. 5 (a) shows a DCES model of our example, where *rest* represents the rest of the treatment process, and *ct* represents the computer tomography. The initial causality in a DCES e.g. $ct \rightarrow rest$ corresponds to the regular path of a process, while the changes carried by modifiers e.g. *cardiac pacemaker* correspond to exceptional one. Other static-causality ESs like a BES and a DES can model the fact that either the computer tomography XOR the X-ray is needed, as shown in Fig. 5(b) and 5(c). The same can be done by an equivalent RCES. However, we argue that none of these models can distinguish between regular and exceptional paths.

Thus our main contributions are: 1. We provide a formal model that allows us to express dynamicity in causality. Using this model, we enhance the PESs yielding SESs, GESs and DCESs. 2. We show the equivalence of SESs and DESs. 3. We show the incomparability of GESs to many other types of ESs. 4. We show that DCESs are strictly more expressive than EBESs and thus strictly more expressive than many other existing types of ESs. 5. We show that DCESs are incomparable to RCESs. 6. The new model succinctly supports modern workflow management systems.

In [5] Crafa et al. defined an Event Structure semantics for the π-calculus based on Prime ESs. Since the latter do not allow for disjunctive causality which they needed, and in order to avoid duplications of events, they extended Prime ESs with a set of bound names, and altered the configuration definition to allow for such disjunction. With Shrinking-Causality ESs—that can express disjunctive causality—this problem could possibly be addressed more naturally without copying events. Here higher-order dynamicity, i.e. to allow for adding and dropping of adders and droppers, might help to deal with the instantiation of variables caused by communications involving bound names.

Up to now, we limit the execution of a DCESs such that an interleaving between adders and droppers of the same causal dependency is forced. As a future work, we want to study the case where modifiers of the same dependency can occur concurrently—read: at the very same instant of time—in DCESs. Similarly, we want to investigate the situation of concurrent occurrence of an adder and its target in GESs. Furthermore, we want to study the ideas of adding and dropping by sets of events or even higher-order dynamics, i.e. events that may change the role of events to adders, dropper or back to normal events. Additionally, the set of possible changes in our newly defined ESs must still be declared statically. We will also investigate the idea that ESs can evolve, by supporting ad hoc changes, such that new dependencies as well as events can be added to a structure.

References

1. Y. Arbach, D. Karcher, K. Peters, and U. Nestmann. Dynamic Causality in Event Structures (Technical Report). Technical report, TU Berlin, 2015. Available at http://arxiv.org/.
2. P. Baldan, N. Busi, A. Corradini, and G.M. Pinna. Domain and event structure semantics for Petri nets with read and inhibitor arcs. *Theoretical Computer Science*, 323(1–3):129–189, 2004.
3. P. Baldan, A. Corradini, and U. Montanari. Contextual Petri Nets, Asymmetric Event Structures, and Processes. *Information and Computation*, 171(1):1–49, 2001.
4. G. Boudol and I. Castellani. Flow Models of Distributed Computations: Three Equivalent Semantics for CCS. *Information and Computation*, 114(2):247–314, 1994.
5. S. Crafa, D. Varacca, and N. Yoshida. Event structure semantics of parallel extrusion in the pi-calculus. In *Proceedings of FoSSaCS*, volume 7213 of *LNCS*, pages 225–239, 2012.
6. J.P. Katoen. *Quantitative and Qualitative Extensions of Event Structures*. PhD thesis, Twente, 1996.
7. D. Kuske and R. Morin. Pomsets for Local Trace Languages. In *Proceedings of CONCUR*, volume 1877 of *LNCS*, pages 426–441. Springer, 2000.
8. R. Langerak. *Transformations and Semantics for LOTOS*. PhD thesis, Twente, 1992.
9. R. Langerak, E. Brinksma, and J.P. Katoen. Causal Ambiguity and Partial Orders in Event Structures. In *Proceedings of CONCUR*, volume 1243 of *LNCS*, pages 317–331. Springer, 1997.
10. M. Reichert, P. Dadam, and T. Bauer. Dealing with forward and backward jumps in workflow management systems. *Software and Systems Modeling*, 2(1):37–58, 2003.
11. A. Rensink. Posets for Configurations! In *Proceedings of CONCUR*, volume 630 of *LNCS*, pages 269–285. Springer, 1992.
12. R. van Glabbeek and G. Plotkin. Event Structures for Resolvable Conflict. In *Proceedings of MFCS*, volume 3153 of *LNCS*, pages 550–561. Springer, 2004.
13. B. Weber, M. Reichert, and S. Rinderle-Ma. Change patterns and change support features - enhancing flexibility in process-aware information systems. *Data & Knowledge Engineering*, 66(3):438 – 466, 2008.
14. G. Winskel. *Events in Computation*. PhD thesis, Edinburgh, 1980.
15. G. Winskel. An introduction to event structures. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, volume 354 of *LNCS*, pages 364–397. Springer, 1989.
16. G. Winskel. Distributed Probabilistic and Quantum Strategies. In *Proceedings of MFPS*, volume 298 of *ENTCS*, pages 403–425. Elsevier, 2013.