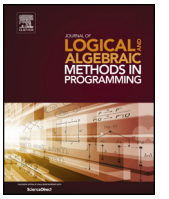
Journal of Logical and Algebraic Methods in Programming 129 (2022) 100807

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp



A semantic model for interacting cyber-physical systems

Benjamin Lion [a,*], Farhad Arbab [a,b], Carolyn Talcott [c]

[a] Leiden University, Leiden, the Netherlands
[b] CWI, Amsterdam, the Netherlands
[c] SRI International, CA, USA

ARTICLE INFO

Article history:
Received 23 January 2022
Received in revised form 22 August 2022
Accepted 22 August 2022
Available online 28 August 2022

Keywords:
Cyber-physical
Interaction
Components
Composition
Algebra

ABSTRACT

We propose a component-based semantic model for Cyber-Physical Systems (CPSs) wherein the notion of a component abstracts the internal details of both cyber and physical processes, to expose a uniform semantic model of their externally observable behaviors expressed as sets of sequences of observations. We introduce algebraic operations on such sequences to model different kinds of component composition. These composition operators yield the externally observable behavior of their resulting composite components through specifications of interactions of the behaviors of their constituent components, as they, e.g., synchronize with or mutually exclude each other's alternative behaviors. Our framework is expressive enough to allow articulation of properties that coordinate desired interactions among composed components within the framework, also as component behavior. We demonstrate the usefulness of our formalism through examples of coordination properties in a CPS consisting of two robots interacting through shared physical resources.

1. Introduction

Compositional approaches in software engineering reduce the complexity of specification, analysis, verification, and construction of software by decomposing a large system into (a) smaller parts, and (b) their interactions. Applied recursively, compositional methods reduce software complexity by breaking the software and its parts into ultimately simple modules, each with a description, properties, and interactions of manageable size. The natural tendency to regard each physical entity as a separate module in a Cyber-Physical System (CPS) makes compositional methods particularly appealing for specification, analysis, verification, and construction of CPSs. However, the distinction between discrete versus continuous transformations in modules representing cyber versus physical processes requires for a semantic model to be sufficiently expressive in order to reflect both discrete and continuous phenomena and capture interactions of cyber-cyber, cyber-physical, and physical-physical pairs of modules. Our work proposes an algebra of components as a semantic model for cyber-physical systems, and is distinguished from existing work in the following three senses.

First, we unify cyber and physical aspects within the same semantic model. While this feature is present in some other works (e.g., signal semantics for cyber-physical systems [1,2], time data streams for connectors [3]), we add a structural constraint by defining an observable as a set of events that happen at the same time. An observation is therefore a pair (O, t) of an observable O occurring at a time t. As a result, we abstract the underlying data flow (e.g., causality rules, input-output ports) that must be implemented for such observables to happen in other models. We believe that this abstraction is different from traditional approaches to design cyber-physical systems and, for instance, may naturally compose a set

* Corresponding author.
E-mail addresses: lion@cwi.nl (B. Lion), farhad@cwi.nl (F. Arbab), carolyn.talcott@gmail.com (C. Talcott).

https://doi.org/10.1016/j.jlamp.2022.100807

of observations occurring at the same time into a new observation formed by the union of their observables. The Timed-Event Stream (TES) model proposed in this paper differs from the trace semantics in [3] in that it explicitly and directly expresses synchronous occurrences of events. Like the one in [3] but unlike many other trace semantics that effectively assume a discrete model of time, the TES model is based on a dense model of time. These distinctions become significant in enabling a compositional semantic model where the sequences of actions of individual components/agents are specified locally, not necessarily in lock-step with those of other entities. It is on this basis that we can define our expressive generic composition operators with interesting algebraic properties. The advantages of this compositional semantics include not just modular, reusable specification of components, but also modular abstractions that allow reasoning and verification following the assume/guarantee methodology.

Second, we make coordination mechanisms explicit and exogenous to components. A component is a standalone entity that exhibits a behavior (which may be described by a finite state automaton, hybrid automaton, or a set of differential equations), and permissible coordinated interactions among components is given by a set of constraints on behaviors of each component. One such composition operation, also widely used in the design of modular systems [3] is set intersection: each component interacts with other components by producing a behavior that is consistent with their shared events. We generalize such composition operations to ease the specification of permissible interactions among cyber-physical components. Each composition operator is parametrized by an interaction signature that captures how two components interact with each other. We also show the benefit in modeling interaction exogenously when it comes to reasoning about, e.g., asymmetric product operations, or proving some algebraic properties. Our approach, in this sense, differs from how control theory models interacting cyber-physical systems, where differences between cyber and physical are first class and lead to, for instance, the use of hybrid models [4]. Instead, we assume that the underlying control is given, and consider the coordination in a discrete framework [5], where (cyber or physical) processes are modeled as components that exhibit sequences of timed-events and relations on those components act as constraints.

Third, we give an alternative view on satisfaction of trace and behavioral properties of cyber-physical systems. We expose properties as components and show how to express coordination as a satisfaction problem (i.e., adding to a system of components a coordinator that restricts each component to a subset of its behavior to comply with a trace property). Moreover, we show that trace properties are not adequate to capture all important properties. We introduce behavioral properties, which are analogous to hyperproperties of [6]. We show, for instance, how the energy adequacy property in a cyber-physical system requires both a behavioral property and a trace property.

More precisely, we introduce an abstract algebra to model interacting cyber-physical systems, where components link through algebraic products to form more complex systems. As a result, a cyber-physical system becomes an algebraic expression (e.g., $C \times_\Sigma P$) whose parts are simpler components (e.g., C, P) and whose products (e.g., $\times_\Sigma$) are parametrized by an interaction signature (e.g., Σ). Throughout the paper, various examples reflect both cyber and physical aspects of components, and a range of interaction signatures is introduced to capture different types of interaction among components. We make explicit the relation between algebraic properties of parametrized products (such as associativity, commutativity, and idempotency) and properties of the interaction signature. As well, we present a co-inductive construction that lifts local constrains on observations to global interaction signatures on components.

Contributions.

- we propose a semantic model of interacting cyber and physical processes based on sequences of observations,
- we define an algebraic framework to express interactions among time sensitive components,
- we give a general mechanism, using a co-inductive construction, to define algebraic operations on components as a lifting of some constraints on observations,
- we introduce two classes of properties on components, trace properties and behavior properties, and demonstrate their application in an example.

This work extends [7] by (1) including a proof for each result, (2) adding several new results about properties of parametrized products on components, and (3) extending the set of examples that use our model.

Our approach differs from more concrete approaches (e.g., operational models, executable specifications, etc.) in the sense that our operations on components model operations of composition at the semantic level.

We first intuitively introduce some key concepts and an example in Section 2. We provide in Section 3 formal definitions for components, their composition, and their properties. We describe a detailed example in Section 4. We present some related work and our future work in Section 5, and conclude the paper in Section 6.

2. Coordination of energy-constrained robots on a field

In this work, we consider a cyber-physical system as a set of interacting processes. Whether a process consists of a physical phenomenon (sun rising, electro-chemical reaction, etc.) or a cyber phenomenon (computation of a function, message exchanges, etc.), it exhibits an externally observable behavior resulting from some internal non-visible actions. Instead of a unified way to describe internals of cyber and physical processes, we propose a uniform description of what we can externally observe of their behavior and interactions.

Table 1
Each column displays a segment of a timed-event stream for a robot, a battery, and a field component, where observables are singleton events. For $t \in \mathbb{R}_+$, we use $R(t)$, $B(t)$, and $F(t)$ to respectively denote the observable at time t for the TES in the Robot, the Battery, and the Field column. An explicit empty set is not mandatory if no event is observed.

	Robot (R)	Battery (B)	Field (F)	Robot-Battery-Field
1 s	$\{(read(loc, R); (0; 0))\}$		$\{(loc(I); (0; 0))\}$	$R(1) \cup F(1)$
2 s	$\{(move(R); (N, 20\text{ W}))\}$	$\{(discharge(B); 20\text{ W})\}$	$\{(move(I); (N, 40\text{ N}))\}$	$R(2) \cup B(2) \cup F(2)$
3 s	$\{(read(loc, R); (0; 1))\}$		$\{(loc(I); (0; 1))\}$	$R(3) \cup F(3)$
4 s	$\{(read(bat, R); 2000\text{ Wh})\}$	$\{(read(B); 2000\text{ Wh})\}$		$R(4) \cup B(4)$
...	...	...	...	...

In this section, we introduce some concepts that we will formalize later. An *event* may describe something like *the sun-rise* or *the temperature reading of* $5°C$. An event occurs at a point in time, yielding an event occurrence (e.g., the sun-rise event occurred at 6:28 am today), and the same event can occur repeatedly at different times (the sun-rise event occurs every day). Typically, multiple events may occur at "the same time" as measured within a measurement tolerance (e.g., the bird vacated the space at the same time as the bullet arrived there; the red car arrived at the middle of the intersection at the same time as the blue car did). We call a set of events that occur together at the same time an *observable*. A pair (O, t) of a set of observable events O together with its time-stamp t represents an *observation*. An observation (O, t) in fact consists of a set of event occurrences: occurrences of events in O at the same time t. We call an infinite sequence of observations a *Timed-Event Stream* (TES). A *behavior* is a set of TESs ranging over a set of events called an *interface*. A *component* is a behavior with an interface.

Consider two robot components, each interacting with its own local battery component, sharing a field resource. The fact that the robots share the field through which they roam, forces them to somehow coordinate their (move) actions. Coordination is a set of constraints imposed on the otherwise possible observable behavior of components. In the case of our robots, if nothing else, at least physics prevents the two robots from occupying the same field space at the same time. More sophisticated coordination may be imposed (by the robots themselves or by some other external entity) to restrict the behavior of the robots and circumvent some undesirable outcomes, including hard constraints imposed by the physics of the field. The behaviors of components consist of timed-event streams, where events may include some measures of physical quantities. We give in the sequel a detailed description of three components, a robot (R), a battery (B), and a field (F), and of their interactions. We use the International System of Units to quantify physical values, with time in seconds (s), charging status in Watt hour (Wh), distance in meters (m), force in newtons (N), speed in meters per second (m s^{-1}).

A *robot* component, with identifier R, has two kinds of events: a read event $(read(bat, R); b)$ that measures the level b of its battery or $(read(loc, R); l)$ that obtains its position l, and a move event $(move(R); (d, \alpha))$ when the robot moves in the direction d with energy α (in W). The TES in the Robot column in Table 1 shows a scenario where robot R reads its location and gets the value $(0; 0)$ at time 1 s, then moves north with 20 W at time 2 s, reads its location and gets $(0; 1)$ at time 3 s, and reads its battery value and gets 2000 Wh at time 4 s,....

A *battery* component, with identifier B, has three kinds of events: a charge event $(charge(B); \eta_c)$, a discharge event $(discharge(B); \eta_d)$, and a read event $(read(B); s)$, where η_d and η_c are respectively the discharge and charge rates of the battery, and s is the current charge status. The TES in the Battery column in Table 1 shows a scenario where the battery discharged at a rate of 20 W at time 2 s, and reported its charge-level of 2000 Wh at time 4 s,....

A *field* component, with identifier F, has two kinds of events: a position event $(loc(I); p)$ that obtains the position p of an object I, and a move event $(move(I); (d, F))$ of the object I in the direction d with traction force F (in N). The TES in the Field column in Table 1 shows a scenario where the field has the object I at location $(0; 0)$ at time 1 s, then the object I moves in the north direction with a traction force of 40 N at time 2 s, subsequently to which the object I is at location $(0; 1)$ at time 3 s,....

When components interact with each other, in a shared environment, behaviors in their composition must also compose with a behavior of the environment. For instance, a battery component may constrain how many amperes it delivers, and therefore restrict the speed of the robot that interacts with it. We specify interaction explicitly as an exogenous binary operation that constrains the composable behaviors of its operand components.

The *robot-battery* interaction imposes that a move event in the behavior of a robot coincides with a discharge event in the behavior of the robot's battery, such that the discharge rate of the battery is proportional to the energy needed by the robot. The physicality of the battery prevents the robot from moving if the energy level of the battery is not sufficient (i.e., such an anomalous TES would not exist in the battery's behavior, and therefore cannot compose with a robot's behavior). Moreover, a read event in the behavior of a robot component should also coincide with a read event in the behavior of its corresponding battery component, such that the two events contain the same charge value.

The *robot-field* interaction imposes that a move event in the behavior of a robot coincides with a move event of an object on the field, such that the traction force on the field is proportional to the energy that the robot put in the move. A read event in the behavior of a robot coincides with a position event of the corresponding robot object on the field, such that the two events contain the same position value. Additional interaction constraints may be imposed by the physics of the field.

For instance, the constraint "no two robots can be observed at the same location" would rule out every behavior where the two robots are observed at the same location.

A TES for the composite Robot-Battery-Field system collects, in sequence, all observations from a TES in a Robot, a Battery, and a Field component behavior, such that at any moment the interaction constraints are satisfied. The column Robot-Battery-Field in Table 1 displays the first elements of such a TES.

3. Components, composition, and properties

The definition of components in this section is similar to the one defined in [3,5]. Intuitively, a component denotes a set of (infinite) sequences of observations. Whether it is a cyber process or a physical process, our notion of component captures all of its possible sequences of observations.

A model of interaction emerges naturally from our component model by relating observation of events from one component to observation of events from another component. Moreover, we give a construction to lift constraints on observations to constraints on infinite sequences of observations, and ultimately define, from those interaction constraints, algebraic operations on components.

3.1. Notations

An *event* is a simplex (the most primitive form of an) observable element. An event may or may not have internal structure. For instance, the successive ticks of a clock are occurrences of a tick event that has no internal structure; successive readings of a thermometer, on the other hand, constitute occurrences of a temperature-reading event, each of which has the internal structure of a name-value pair. Similarly, we can consider successive transmissions by a mobile sensor as occurrences of a structured event, each instance of which includes geolocation coordinates, barometric pressure, temperature, humidity, etc. Regardless of whether or not events have internal structures, in the sequel, we regard events as uninterpreted simplex observable elements.

Notation 1 *(Events).* We use $\mathbb{E}$ to denote the universal set of events.

An *observable* is a set of event occurrences that happen together and an *observation* is a pair (O, t) of an observable O and a time-stamp $t \in \mathbb{R}_+$.[1] An observation (O, t) represents an act of atomically observing occurrences of events in O at time t. Atomically observing occurrences of events in O at time t means there exists a small $\epsilon \in \mathbb{R}_+$ such that during the time interval $[t-\epsilon, t+\epsilon]$:

1. every event $e \in O$ is observed exactly once,[2] and
2. no event $e \notin O$ is observed.

We write $\langle s_0, s_1, ..., s_{n-1}\rangle$ to denote a *finite sequence of size n* of elements over an arbitrary set S, where $s_i \in S$ for $0 \leq i \leq n-1$. The set of all finite sequences of elements in S is denoted as S^*. A *stream* over a domain S is a function $\sigma : \mathbb{N} \to S$.[3] We use $\sigma(i)$ to represent the $i+1^{st}$ element of σ, and given a finite sequence $s = \langle s_0, ..., s_{n-1}\rangle$, we write $s \cdot \sigma$ to denote the stream $\tau \in \mathbb{N} \to S$ such that $\tau(i) = s_i$ for $0 \leq i \leq n-1$ and $\tau(i) = \sigma(i-n)$ for $n \leq i$. We use $\sigma^{(n)}$ to denote the *n-th* derivative of σ, such that $\sigma^{(n)}(i) = \sigma(i+n)$ for all $i \in \mathbb{N}$. We use σ' as an abbreviation for the first derivative of the stream σ, i.e., $\sigma' = \sigma^{(1)}$. We use $\mathcal{P}(X)$ to denote the power set of X.

A *Timed-Event Stream (TES)* over a set of events E and a set of time-stamps $\mathbb{R}_+$ is a stream $\sigma \in \mathbb{N} \to (\mathcal{P}(E) \times \mathbb{R}_+)$ where, for every $i \in \mathbb{N}$, let $\sigma(i) = (O_i, t_i)$ and:

1. $t_i < t_{i+1}$, [i.e., time monotonically increases] and
2. for every $n \in \mathbb{N}$, there exists $k \in \mathbb{N}$ such that $t_k > n$ [i.e., time is non-Zeno progressive].

Notation 2 *(Time stream).* We use $OS(\mathbb{R}_+)$ to refer to the set of all monotonically increasing and non-Zeno infinite sequences of elements in $\mathbb{R}_+$.

Notation 3 *(Timed-Event Stream).* We use $TES(E)$ to denote the set of all TESs whose observables are subsets of the event set E with elements in $\mathbb{R}_+$ as their time-stamps.

[1] Any totally ordered dense set would be suitable as the domain for time (e.g., positive rationals $\mathbb{Q}_+$). For simplicity, we use $\mathbb{R}_+$, the set of real numbers $r \geq 0$ for this purpose.

[2] A finer time granularity, i.e., a smaller ϵ, may reveal some ordering relation on the set of events that occur in the same set of observation.

[3] The set $\mathbb{N}$ denotes the set of natural numbers $n \geq 0$.

Given a sequence $\sigma \in TES(E)$ with $\sigma(i) = (O_i, t_i)$ for $i \in \mathbb{N}$, we use the projections $\mathsf{pr}_1(\sigma) \in \mathbb{N} \to \mathcal{P}(E)$ and $\mathsf{pr}_2(\sigma) \in OS(\mathbb{R}_+)$ to denote respectively the sequence of observables where $\mathsf{pr}_1(\sigma)(i) = O_i$ and the sequence of time stamps where $\mathsf{pr}_2(\sigma)(i) = t_i$.

Notation 4 *(Observable time).* For $\sigma \in TES(E)$ and $t \in \mathbb{R}_+$, we use $\sigma(t)$ to denote the observable O in σ if there exists $i \in \mathbb{N}$ with $\sigma(i) = (O, t)$, and $\emptyset$ otherwise. We write $\Theta(\sigma)$ for the set of all $t \in \mathbb{R}_+$ such that there exists $i \in \mathbb{N}$ with $\sigma(i) = (O_i, t)$ with $O_i \subseteq E$.

Note that, for $t \in \mathbb{R}_+$ where $\sigma(t) = \emptyset$, the meaning of $\sigma(t)$ is ambiguous as it may mean either $t \notin \Theta(\sigma)$, or there exists an $i \in N$ such that $\sigma(i) = (\emptyset, t)$. The ambiguity is resolved by checking if $t \in \Theta(\sigma)$.

Notation 5 *(Pair derivative).* For a pair (σ, τ) of TESs, we use $(\sigma, \tau)'$ to denote the new pair of TESs for which the observation(s) with the smallest time stamp has been dropped, i.e., $(\sigma, \tau)' = (\sigma^{(x)}, \tau^{(y)})$ with x (resp. y) is 1 if $\mathsf{pr}_2(\sigma)(0) \leq \mathsf{pr}_2(\tau)(0)$ (resp. $\mathsf{pr}_2(\tau)(0) \leq \mathsf{pr}_2(\sigma)(0)$) and 0 otherwise.

3.2. Components

The design of complex systems becomes simpler if such systems can be decomposed into smaller sub-systems that interact with each other. In order to simplify the design of cyber-physical systems, we abstract from the internal details of both cyber and physical processes, to expose a uniform semantic model. As a first class entity, a component encapsulates a behavior (set of TESs) and an interface (set of events).

Like existing semantic models, such as time-data streams [3], time signal [2], or discrete clock [8], we use a dense model of time. However, we allow for arbitrary but finite interleavings of observations. In addition, our structure of an observation imposes atomicity of event occurrences within an observation. These distinctions mean that for every $\sigma(i) = (O, t), i \geq 0$ of a $\sigma \in TES(E)$: (1) O is finite; and (2) there exists a real number $\epsilon > 0$ such that in the open interval $(t - \epsilon, t + \epsilon)$ no event $e \notin O$ occurs, and every event $e \in O$ occurs exactly once. Such a constraint abstracts from the precise timing of the occurrence of each event in the set O, and turns an observation into an all-or-nothing transaction.

Definition 1 *(Component).* A component is a tuple $C = (E, L)$ where $E \subseteq \mathbb{E}$ is a set of events, and $L \subseteq TES(E)$ is a set of TESs. We call E the *interface* and L the externally observable *behavior* of C.

In contrast with other component models where observables range over the same universal set of events, therefore making component overly specified, our model encapsulates the set of observable events of a component in its interface. Thus, a component *cannot observe* an event that is not in its interface. Moreover, Definition 1 makes no distinction between cyber and physical components. We use the following examples to describe some cyber and physical aspects of components.

Example 1. Consider a set of two events $E = \{0, 1\}$, and restrict our observations to $\{1\}$ and $\{0\}$. A component whose behavior contains TESs with alternating observations of $\{1\}$ and $\{0\}$ is defined by the tuple (E, L) where

$$L = \{\sigma \in TES(E) \mid \forall i \in \mathbb{N}. \quad (\mathsf{pr}_1(\sigma)(i) = \{0\} \wedge \mathsf{pr}_1(\sigma)(i+1) = \{1\}) \vee (\mathsf{pr}_1(\sigma)(i) = \{1\} \wedge \mathsf{pr}_1(\sigma)(i+1) = \{0\})\}$$

Note that this component is oblivious to time, and any stream of monotonically increasing non-Zeno real numbers would serve as a valid stream of time stamps for any such sequence of observations. ■

Example 2. Consider a component encapsulating a continuous function $f : (D_0 \times \mathbb{R}_+) \to D$, where D_0 is a set of initial values, and D is the codomain of values for f. Such a function can describe the evolution of a physical system over time, where $f(d_0, t) = d$ means that at time t the state of the system is described by the value $d \in D$ if initialized with d_0. We define the set of all events for this component as the range of function f given an initial parameter $d_0 \in D_0$. The component is then defined as the pair (D, L_f) such that:

$$L_f = \{\sigma \in TES(D) \mid \exists d_0 \in D_0. \, \forall i \in \mathbb{N}. \, \mathsf{pr}_1(\sigma)(i) = \{f(d_0, \mathsf{pr}_2(\sigma)(i))\}\}$$

Observe that the behavior of this component contains all possible discrete samplings of the function f at monotonically increasing and non-Zeno sequences of time stamp. Different instances of f would account for various cyber and physical aspects of components. ■

3.3. Composition

A complex system typically consists of multiple components that interact with each other. The example in Section 2 shows three components, a *robot*, a *battery*, and a *field*, where, for instance, a move observable of a robot must coincide

with an accommodating move observable of the field and a discharge observable of its battery. The design challenge is to faithfully represent the interactions among involved components, while keeping the description modular, i.e., specify the robot, the battery, and the field as separate, independent, but interacting components. For that purpose, we capture in an interaction signature the type of the interaction between a pair of components, and we define a family of binary products acting on components, each parametrized with an interaction signature. As a result, the product of two components, under a given interaction signature, returns a new component whose behavior reflects that the two operand components joint behavior is constrained according to the interaction signature. Such construction opens possibilities for modular reasoning both about the interaction among components and about their resulting composite behavior.

An interaction signature consists of two elements: a composability relation and a composition function. The composability relation specifies which pairs[4] of TESs are allowed to compose, and the composition function constructs a new TES out of a pair TESs. The condition for two TESs to be composable may depend on an external context. For instance, the observation of event a at time t in a TES may conflict with the observation of event b at that same time t in another TES in a context where the latter could have observed a as well. To capture this notion, we generalized the notion of a composability relation to take as parameter a pair of carrier sets of events that acts as a context of alternative events for the pair of TESs. Then, when we write $(\sigma, \tau) \in R(E_1, E_2)$, we mean that σ and τ are composable under the composability relation R given their respective context E_1 and E_2.

Definition 2 *(Composability relation on TESs).* A composability relation is a parametrized relation R such that for all $E_1, E_2 \subseteq \mathbb{E}$, we have $R(E_1, E_2) \subseteq \mathit{TES}(E_1) \times \mathit{TES}(E_2)$.

Definition 3 *(Symmetry).* A parametrized relation Q is *symmetric* if, for all (x_1, x_2) and for all (X_1, X_2): $(x_1, x_2) \in Q(X_1, X_2) \iff (x_2, x_1) \in Q(X_2, X_1)$.

A composability relation on TESs serves as a necessary constraint for two TESs to compose. We define *composition* of TESs as the act of forming a new TES out of two TESs.

Definition 4. A composition function $\oplus$ on TES is a function $\oplus : \mathit{TES}(\mathbb{E}) \times \mathit{TES}(\mathbb{E}) \to \mathit{TES}(\mathbb{E})$.

In order to simplify the development of the theory of components, we group a pair of a composability relation and a composition function into an *interaction signature*.

Definition 5. An interaction signature $\Sigma = (R, \oplus)$ is a pair of a composability relation R and a composition function $\oplus$.

Example 3 *(Union of TESs).* The operation $\cup$ forms the interleaved union of observables occurring in a pair of TESs, i.e., for two TESs σ and τ, we define $\sigma \cup \tau$ to be the TES such that $\Theta(\sigma \cup \tau) = \Theta(\sigma) \cup \Theta(\tau)$ and $(\sigma \cup \tau)(t) = \sigma(t) \cup \tau(t)$ for all $t \in \Theta(\sigma) \cup \Theta(\tau)$. ■

The following examples present some useful interaction signatures for composition of TESs that, e.g., enforce synchronization or mutual exclusion of observables.

Example 4 *(Synchronous interaction).* The composability relation $\bowtie(E_1, E_2)$ forces observations of events shared by E_1 and E_2 to happen at the same time, i.e., $(\sigma, \tau) \in \bowtie(E_1, E_2)$ if and only if $\sigma(t) \cap E_2 = \tau(t) \cap E_1$ for all $t \in \Theta(\sigma) \cup \Theta(\tau)$. A synchronous interaction signature $\Sigma = (\bowtie, \cup)$ filters pairs of TESs that satisfy the $\bowtie$ relation and merges composable pairs of observations. ■

Example 5 *(Asynchronous interaction).* The composability relation $\nparallel(E_1, E_2)$ prevents observations of events shared by E_1 and E_2 from happening at the same time, i.e., $(\sigma, \tau) \in \nparallel(E_1, E_2)$ if and only if $\sigma(t) \cap \tau(t) = \emptyset$ for all $t \in \Theta(\sigma) \cup \Theta(\tau)$. An asynchronous interaction signature $\Sigma = (\nparallel, \cup)$ filters pairs of TESs that satisfy the $\nparallel$ relation and merges composable pairs. ■

Example 6 *(Free interaction).* A free interaction signature, $\Sigma = (\top, \cup)$, uses $\top$ for the most permissive composability relation on TESs such that, for any $E_1, E_2 \subseteq \mathbb{E}$ and any $\sigma \in \mathit{TES}(E_1)$ and $\tau \in \mathit{TES}(E_2)$, we have $(\sigma, \tau) \in \top(E_1, E_2)$. ■

We define a binary product operation on components, parametrized by an interaction signature. Intuitively, the newly formed component describes, by its behavior, the evolution of the joint system under the constraint that the interactions in the system satisfy the composability relation. Formally, the product operation returns another component, whose set of

[4] Non-binary relations may also be considered, i.e., constraints imposed on more than two components.

events is the union of sets of events of its operands, and its behavior is obtained by composing all pairs of TESs in the behavior of its operands deemed composable by the composability relation.

Definition 6 *(Product).* Let $\Sigma = (R, \oplus)$ be an interaction signature, and $C_i = (E_i, L_i)$, $i \in \{1, 2\}$, two components. The product of C_1 and C_2, under Σ, denoted as $C_1 \times_\Sigma C_2$, is the component (E, L) where $E = E_1 \cup E_2$ and L is defined by

$$L = \{\sigma_1 \oplus \sigma_2 \mid \sigma_1 \in L_1,\ \sigma_2 \in L_2,\ (\sigma_1, \sigma_2) \in R(E_1, E_2)\}$$

The following examples define several products on components given the interaction signatures introduced in Example 4, 5, and 6.

Example 7 *(Synchronous product).* The behavior of component $C_1 \times_{(\bowtie, \oplus)} C_2$ contains TESs obtained from the composition under $\oplus$ of every pair $\sigma_1 \in L_1$ and $\sigma_2 \in L_2$ of TESs that are related by the synchronous composability relation $\bowtie$ (see Example 4) which excludes all event occurrences that do not synchronize. ■

Example 8 *(Asynchronous product).* The behavior of component $C_1 \times_{(\nparallel, \oplus)} C_2$ contains TESs resulting from the composition under $\oplus$ of every pair $\sigma_1 \in L_1$ and $\sigma_2 \in L_2$ of TESs that are related by the mutual exclusion composability relation $\nparallel$ (see Example 5) which may exclude some simultaneous event occurrences. ■

Example 9 *(Free product).* The behavior of component $C_1 \times_{(\top, \oplus)} C_2$ contains every TES obtained from the composition under $\oplus$ of every pair $\sigma_1 \in L_1$ and $\sigma_2 \in L_2$ of TESs. This product does not impose any constraint on event occurrences of its operands (see Example 6). ■

Example 10. Consider a suitable interaction signature Σ that captures the interactions between a robot R and its field F, such that the expression $R \times_\Sigma F$ represents the resulting system. For instance, Σ may force every observable move of the robot to synchronize with a displacement of the robot on the field F, and every read observable of the robot with a location displayed by the field. In the case of two interacting robots roaming on the same field, one would like to build the resulting system compositionally as an expression of the form $(R_1 \times_{\Sigma_1} F_1) \times_{\Sigma_3} (R_2 \times_{\Sigma_2} F_2)$, where each of the Σ_i locally captures the interaction between its respective component. Note that, for instance, Σ_3 may enforce that the two fields F_1 and F_2 exclude the joint observations of R_1 and R_2 to be at the same location. ■

The product of two components indirectly depends on the interface of its operands, since its composability relation does so. Therefore, it is *a priori* not certain that algebraic properties such as commutativity or associativity hold for such user defined products. Algebraic properties are important when designing a complex system in order to find equivalent and sometimes simpler expressions. Lemma 1 relates properties of a parametrized product with properties of its parameter, i.e., properties of the interaction signature. Intuitively, the first item of Lemma 1 considers interaction signatures that yield symmetric operations. As a result, the order of which components appear in the product parametrized by such signatures is irrelevant. The second item shows conditions on interaction signatures that allow flattening of nested products: the product of A with $B \times_\Sigma C$ becomes equivalent to the product of $A \times_\Sigma B$ with C. When an interaction signature satisfies both algebraic properties, the resulting product acts as an n-ary top level operator on a multiset of components. For instance, the synchronous interaction signature of Example 4 is one such top level n-ary operator.[5]

Lemma 1. *Let $\Sigma = (R, \oplus)$ be an interaction signature. Then:*

- *if R is symmetric, then $\times_\Sigma$ is* commutative *if and only if $\sigma_1 \oplus \sigma_2 = \sigma_2 \oplus \sigma_1$ for all $(\sigma_1, \sigma_2) \in R$;*
- *if R is such that, for all $E_1, E_2, E_3 \subseteq \mathbb{E}$,*

  $$(\sigma_1, \sigma_2 \oplus \sigma_3) \in R(E_1, E_2 \cup E_3) \wedge (\sigma_2, \sigma_3) \in R(E_2, E_3) \iff (\sigma_1, \sigma_2) \in R(E_1, E_2) \wedge (\sigma_1 \oplus \sigma_2, \sigma_3) \in R(E_1 \cup E_2, E_3)$$

  then $\times_\Sigma$ is associative *if and only if $\sigma_1 \oplus (\sigma_2 \oplus \sigma_3) = (\sigma_1 \oplus \sigma_2) \oplus \sigma_3$ for all $(\sigma_1, \sigma_2 \oplus \sigma_3) \in R(E_1, E_2 \cup E_3)$ with $(\sigma_2, \sigma_3) \in R(E_2, E_3)$;*
- *if for all $E \subseteq \mathbb{E}$ and $\sigma, \tau \in TES(E)$, we have $(\sigma, \tau) \in R(E, E) \implies \sigma = \tau$, then $\times_\Sigma$ is* idempotent *if and only if $\sigma \oplus \sigma = \sigma$ for all $(\sigma, \sigma) \in R$.*

Proof. Commutativity. Let $C_1 = (E_1, L_1)$ and $C_2 = (E_2, L_2)$ be two components, and $\Sigma = (R, \oplus)$ be an interaction signature with R symmetric as in Definition 3. We write $C = (E, L) = C_1 \times_\Sigma C_2$ and $C' = (E', L') = C_2 \times_\Sigma C_1$. We first observe that

[5] Distributivity holds for some products. We leave the study of the conditions under which distributivity holds as future work.

$E = E_1 \cup E_2 = E'$. The condition for the product of two components to be commutative reduces to showing that $L = L'$, also equivalently written as:

$$\begin{aligned} L = L' \iff & \{\sigma_1 \oplus \sigma_2 \mid \sigma_1 \in L_1,\ \sigma_2 \in L_2,\ (\sigma_1, \sigma_2) \in R(E_1, E_2)\} \\ & = \{\sigma_2 \oplus \sigma_1 \mid \sigma_1 \in L_1,\ \sigma_2 \in L_2,\ (\sigma_2, \sigma_1) \in R(E_2, E_1)\} \end{aligned}$$

If $\sigma_1 \oplus \sigma_2 = \sigma_2 \oplus \sigma_1$ for $(\sigma_1, \sigma_2) \in R(E_1, E_2)$, then $L = L'$ and $\times_\Sigma$ is commutative.

Oppositely, if $L = L'$, we show that $\oplus$ is commutative. Let C_σ be the component $(E_\sigma, \{\sigma\})$ where $E_\sigma = \bigcup\{\sigma(i) \mid i \in \mathbb{N}\}$. Thus, for any $(\sigma_1, \sigma_2) \in R(E_1, E_2)$, $C_{\sigma_1} \times_\Sigma C_{\sigma_2} = (E_{\sigma_1} \cup E_{\sigma_2}, \{\sigma_1 \oplus \sigma_2\}$. A necessary condition for $\times_\Sigma$ to be commutative is that $\{\sigma_1 \oplus \sigma_2\} = \{\sigma_2 \oplus \sigma_1\}$, which imposes that $\sigma_1 \oplus \sigma_2 = \sigma_2 \oplus \sigma_1$.

Associativity. Let $(R, \oplus)$ be a pair of a composability relation and a composition function on TESs with R such that, for every $(\sigma_1, \sigma_2, \sigma_3) \in L_1 \times L_2 \times L_3$:

$$\begin{aligned} & (\sigma_1, \sigma_2) \in R(E_1, E_2) \wedge (\sigma_1 \oplus \sigma_2, \sigma_3) \in R(E_1 \cup E_2, E_3) \iff \\ & (\sigma_2, \sigma_3) \in R(E_2, E_3) \wedge (\sigma_1, \sigma_2 \oplus \sigma_3) \in R(E_1, E_2 \cup E_3) \end{aligned}$$

We consider three components $C_i = (E_i, L_i)$, with $i \in \{1, 2, 3\}$.

The set of events for component $((C_1 \times_\Sigma C_2) \times_\Sigma C_3)$ is the set $E_1 \cup E_2 \cup E_3$, which is equal to the set of events for component $(C_1 \times_\Sigma (C_2 \times_\Sigma C_3))$.

Let L' and L'' respectively be the behaviors of components $(C_1 \times_\Sigma C_2) \times_\Sigma C_3$ and $C_1 \times_\Sigma (C_2 \times_\Sigma C_3)$. If $\sigma_1 \oplus (\sigma_2 \oplus \sigma_3) = (\sigma_1 \oplus \sigma_2) \oplus \sigma_3$ for all $(\sigma_1, \sigma_2 \oplus \sigma_3) \in R(E_1, E_2 \cup E_3)$ with $(\sigma_2, \sigma_3) \in R(E_2, E_3)$, then $L' = L''$. We show some sufficient conditions for $L' = L''$, also written as

$$\begin{aligned} L' & = \{(\sigma_1 \oplus \sigma_2) \oplus \sigma_3 \mid \sigma_1 \in L_1,\ \sigma_2 \in L_2,\ \sigma_3 \in L_3.\ (\sigma_1, \sigma_2) \in R(E_1, E_2) \wedge (\sigma_1 \oplus \sigma_2, \sigma_3) \in R(E_1 \cup E_2, E_3)\} \\ & = \{(\sigma_1 \oplus \sigma_2) \oplus \sigma_3 \mid \sigma_1 \in L_1,\ \sigma_2 \in L_2,\ \sigma_3 \in L_3.\ (\sigma_2, \sigma_3) \in R(E_2, E_3) \wedge (\sigma_1, \sigma_2 \oplus \sigma_3) \in R(E_1, E_2 \cup E_3)\} \\ & = \{\sigma_1 \oplus (\sigma_2 \oplus \sigma_3) \mid \sigma_1 \in L_1,\ \sigma_2 \in L_2,\ \sigma_3 \in L_3.\ (\sigma_2, \sigma_3) \in R(E_2, E_3) \wedge (\sigma_1, \sigma_2 \oplus \sigma_3) \in R(E_1, E_2 \cup E_3)\} \\ & = L'' \end{aligned}$$

using the assumption on R for the first equality, and the assumption on $\oplus$ for the second equality.

Let $(\sigma_1, \sigma_2 \oplus \sigma_3) \in R(E_1, E_2 \cup E_3)$ with $(\sigma_2, \sigma_3) \in R(E_2, E_3)$, then $C_{\sigma_1} \times_\Sigma (C_{\sigma_2} \times_\Sigma C_{\sigma_3}) = (C_{\sigma_1} \times_\Sigma C_{\sigma_2}) \times_\Sigma C_{\sigma_3}$ which then implies that $\sigma_1 \oplus (\sigma_2 \oplus \sigma_3) = (\sigma_1 \oplus \sigma_2) \oplus \sigma_3$.

Idempotency. We show that if for all $E \subseteq \mathbb{E}$, and $\sigma, \tau \in TES(E)$, we have that $(\sigma, \tau) \in R(E, E)$ implies $\sigma = \tau$, then $\times_\Sigma$ is idempotent if and only if $\sigma \oplus \sigma = \sigma$ for $(\sigma, \sigma) \in R(E, E)$. We first observe that, given a component $C = (E, L)$, the component $C \times_\Sigma C = (E, L')$ has the same set of events, E.

We show that $(\sigma_1, \sigma_2) \in R(E, E) \implies \sigma_1 = \sigma_2$ and $\oplus$ idempotent is a sufficient condition for having $L' = L$. Indeed,

$$\begin{aligned} L' & = \{\sigma_1 \oplus \sigma_2 \mid \sigma_1, \sigma_2 \in L,\ (\sigma_1, \sigma_2) \in R(E, E)\} \\ & = \{\sigma_1 \oplus \sigma_1 \mid \sigma_1 \in L\} \\ & = L \end{aligned}$$

Similar to the previous cases, if for all $E \subseteq \mathbb{E}$, and $\sigma, \tau \in TES(E)$, we have that $(\sigma, \tau) \in R(E, E)$ implies $\sigma = \tau$, then $\times_\Sigma$ is idempotent if and only if $\oplus$ is idempotent. Conversely, if $C_\sigma \times_\Sigma C_\sigma = C_\sigma$ and $(\sigma, \sigma) \in R(E, E)$, then $\sigma \oplus \sigma = \sigma$. □

The algebraic nature of our formalism allows the possibility to introduce other kinds of operations on components, such as division. Intuitively, the operation of division is parametrized by an interaction signature Σ and follows two steps. First, the set of quotients of component A divided by component B is constructed as the set of all components C such that $A = B \times_\Sigma C$. Practically, every element in the set of quotients leads to the same composite behavior captured in A, when composed with B under Σ. Then, one component is chosen from the set of quotients as the result of division. We leave as future work the study of the structure of the set of all quotients and the choice of a specific element from that set to define the operation of division.

3.4. A co-inductive construction for composition operators

In this section, we show how local constraints on observations can be co-inductively *lifted* to global constraints on TESs. We get, as a result, a finite specification of some interaction signatures using simpler relations on observations. Moreover, we get a co-inductive proof mechanism to relate an interaction signature defined on TESs with an interaction signature lifted from constraints on observables, as shown in Lemma 4. Such construction gives, as well, an operational perspective on deriving an interaction signature as a step-wise constraint imposed on observables. Practically, the operational approach of the co-inductive definition is relevant when considering robots and their step-wise decision on their next observation.

The intuition for such construction is that, in some cases, the condition for two TESs to be composable depends only on a composability relation on observations. An example of composability constraint for a robot with its battery and a field enforces that each *move* event *discharges* the battery and *changes* the state of the field. As a result, every *move* event observed by the robot must coincide with a *discharge* event observed by the battery and a change of state observed by the field. The lifting of such composability relation on observations to a constraint on TESs is defined co-inductively. Finally, Lemma 11 gives weaker conditions for Lemma 1 to hold.

Definition 7 *(Composability relation on observations).* A composability relation on observations is a parametrized relation κ such that for all pairs $(E_1, E_2) \in \mathcal{P}(\mathbb{E}) \times \mathcal{P}(\mathbb{E})$, we have $\kappa(E_1, E_2) \subseteq (\mathcal{P}(E_1) \times \mathbb{R}_+) \times (\mathcal{P}(E_2) \times \mathbb{R}_+)$.

The following examples define locally on observations some relations analogous to those defined globally on TESs in Example 4 and Example 5.

Example 11. We give two examples of composability relations on observations:

- $((O_1, t_1), (O_2, t_2)) \in \kappa^{sync}(E_1, E_2)$ if and only if every shared event always occurs at the same time, i.e., $t_1 < t_2$ implies $O_1 \cap E_2 = \emptyset$, and $t_2 < t_1$ implies $O_2 \cap E_1 = \emptyset$, and $t_2 = t_1$ implies $O_1 \cap E_2 = O_2 \cap E_1$;
- $((O_1, t_1), (O_2, t_2)) \in \kappa^{async}(E_1, E_2)$ if and only if no shared event occurs at the same time, i.e., $t_1 = t_2$ implies $O_1 \cap E_2 = \emptyset = O_2 \cap E_2$. ■

For two composability relations κ_1, κ_2, their intersection or union, written $\kappa_1 \cap \kappa_2$ and $\kappa_1 \cup \kappa_2$ respectively, is defined, for any $E_1, E_2, E_3 \subseteq \mathbb{E}$, as $(\kappa_1 \cap \kappa_2)(E_1, E_2) = \kappa_1(E_1, E_2) \cap \kappa_2(E_1, E_2)$ and $(\kappa_1 \cup \kappa_2)(E_1, E_2) = \kappa_1(E_1, E_2) \cup \kappa_2(E_1, E_2)$.

Definition 8 *(Lifting - composability relation).* Let κ be a composability relation on observations, and, for any $\mathcal{R} \subseteq TES(E_1) \times TES(E_2)$, let $\Phi_\kappa(E_1, E_2)(\mathcal{R}) \subseteq TES(E_1) \times TES(E_2)$ be such that:

$$\Phi_\kappa(E_1, E_2)(\mathcal{R}) = \{(\tau_1, \tau_2) \mid (\tau_1(0), \tau_2(0)) \in \kappa(E_1, E_2) \wedge (\tau_1, \tau_2)' \in \mathcal{R})\}$$

The *lifting* of κ on TESs, written $[\kappa]$, is the parametrized relation obtained by taking the greatest post fixed point of the function $\Phi_\kappa(E_1, E_2)$ for arbitrary pair $E_1, E_2 \subseteq \mathbb{E}$, i.e., the relation $[\kappa](E_1, E_2) = \bigcup_{\mathcal{R} \subseteq TES(E_1) \times TES(E_2)} \{\mathcal{R} \mid \mathcal{R} \subseteq \Phi_\kappa(E_1, E_2)(\mathcal{R})\}$.

Lemma 2 *(Correctness of lifting). For any $E_1, E_2 \subseteq \mathbb{E}$, the function $\Phi_\kappa(E_1, E_2)$ is monotone, and therefore has a greatest post fixed point.*

Proof. Let κ be a composability relation on observations, and let $E_1, E_2 \subseteq \mathbb{E}$. We recall that the function $\Phi_\kappa(E_1, E_2)$ is such that, for any $\mathcal{R} \subseteq TES(E_1) \times TES(E_2)$:

$$\Phi_\kappa(E_1, E_2)(\mathcal{R}) = \{(\tau_1, \tau_2) \mid \quad (\tau_1(0), \tau_2(0)) \in \kappa(E_1, E_2) \wedge (\tau_1, \tau_2)' \in \mathcal{R})\}$$

Let $\mathcal{R}_1, \mathcal{R}_2 \subseteq TES(E_1) \times TES(E_2)$ be such that $\mathcal{R}_1 \subseteq \mathcal{R}_2$. We show that $\Phi_\kappa(E_1, E_2)(\mathcal{R}_1) \subseteq \Phi_\kappa(E_1, E_2)(\mathcal{R}_2)$. For any $(\tau_1, \tau_2) \in TES(E_1) \times TES(E_2)$,

$$\begin{aligned}(\tau_1, \tau_2) \in \Phi_\kappa(E_1, E_2)(\mathcal{R}_1) &\iff (\tau_1(0), \tau_2(0)) \in \kappa(E_1, E_2) \wedge (\tau_1, \tau_2)' \in \mathcal{R}_1) \\ &\implies (\tau_1(0), \tau_2(0)) \in \kappa(E_1, E_2) \wedge (\tau_1, \tau_2)' \in \mathcal{R}_2) \\ &\implies (\tau_1, \tau_2) \in \Phi_\kappa(E_1, E_2)(\mathcal{R}_2)\end{aligned}$$

Therefore, $\mathcal{R}_1 \subseteq \mathcal{R}_2$ implies that $\Phi_\kappa(E_1, E_2)(\mathcal{R}_1) \subseteq \Phi_\kappa(E_1, E_2)(\mathcal{R}_2)$, and we conclude that $\Phi_\kappa(E_1, E_2)$ is monotonic. We use the Knaster-Tarski theorem, where the underlying lattice is the powerset of TESs with inclusion relation, for the existence of a greatest fixed point of the monotonic function $\Phi_\kappa(E_1, E_2)$ applying to that lattice. Thus, $\Phi_\kappa(E_1, E_2)$ has a greatest fixed point defined as:

$$[\kappa](E_1, E_2) = \bigcup \{\mathcal{R} \mid \mathcal{R} \subseteq \Phi_\kappa(E_1, E_2)(\mathcal{R})\} \quad \square$$

Lemma 3. *If κ is a composability relation on observations, then the lifting $[\kappa]$ is a composability relation on TESs. Moreover, if κ is symmetric (as in Definition 3), then $[\kappa]$ is symmetric.*

Proof. We first note that, given a composability relation κ on observables, the lifting $[\kappa]$ is a composability relation on TESs. Indeed, for any pair of interfaces $E_1, E_2 \subseteq \mathbb{E}$, any $(\sigma, \tau) \in [\kappa](E_1, E_2)$ is a pair in $TES(E_1) \times TES(E_2)$.

If κ is symmetric (as in Definition 3), we show that $[\kappa]$ is also symmetric. Given a set $\mathcal{R} \subseteq TES(E_1) \times TES(E_2)$, we use the notation $\overline{\mathcal{R}}$ to denote the smallest set such that $(\sigma, \tau) \in \mathcal{R} \iff (\tau, \sigma) \in \overline{\mathcal{R}}$. Let $E_1, E_2 \subseteq \mathbb{E}$.

If κ is symmetric, then for $\mathcal{R} \subseteq TES(E_1) \times TES(E_2)$,

$$\begin{aligned}\Phi_\kappa(E_1,E_2)(\mathcal{R}) &= \{(\tau_1,\tau_2) \mid \quad (\tau_1(0),\tau_2(0)) \in \kappa(E_1,E_2) \wedge (\tau_1,\tau_2)' \in \mathcal{R})\} \\ &= \{(\tau_1,\tau_2) \mid \quad (\tau_2(0),\tau_1(0)) \in \kappa(E_2,E_1) \wedge (\tau_1,\tau_2)' \in \mathcal{R})\} \\ &= \{(\tau_1,\tau_2) \mid \quad (\tau_2(0),\tau_1(0)) \in \kappa(E_2,E_1) \wedge (\tau_2,\tau_1)' \in \overline{\mathcal{R}})\} \\ &= \{(\tau_1,\tau_2) \mid \quad (\tau_2,\tau_1) \in \Phi_\kappa(E_2,E_1)(\overline{\mathcal{R}})\} \quad (1)\end{aligned}$$

which shows that $[\kappa]$ is symmetric since, for any $E_1, E_2 \subseteq \mathbb{E}$, $[\kappa](E_1,E_2) = \bigcup_{\mathcal{R} \subseteq TES(E_1)\times TES(E_2)} \{\mathcal{R} \mid \mathcal{R} \subseteq \Phi_\kappa(E_1,E_2)(\mathcal{R})\}$, and

$$\begin{aligned}(\sigma,\tau) \in [\kappa](E_1,E_2) &\iff \exists \mathcal{R}.\ (\sigma,\tau) \in \mathcal{R} \wedge \mathcal{R} \subseteq \Phi_\kappa(E_1,E_2)(\mathcal{R}) \\ &\iff \exists \overline{\mathcal{R}}.\ (\tau,\sigma) \in \overline{\mathcal{R}} \wedge \overline{\mathcal{R}} \subseteq \Phi_\kappa(E_2,E_1)(\overline{\mathcal{R}}) \\ &\iff (\tau,\sigma) \in [\kappa](E_2,E_1)\end{aligned}$$

where the first equivalence is given by the fact that $[\kappa](E_1,E_2)$ is the greatest post fixed point of $\Phi_\kappa(E_1,E_2)$, the second equivalence is obtained from equality (1), and the third equivalence is given by the fact that $[\kappa](E_2,E_1)$ is the greatest post fixed point. □

As a consequence of Lemma 3, any composability relation on observations gives rise to a composability relation on TESs. We give two examples that construct co-inductively the composability relation on TESs from a composability relation on observations. For the following definitions, let $C_1 = (E_1, L_1)$ and $C_2 = (E_2, L_2)$ be two components, and $\oplus$ be a composition function on TESs. We use $\sqcap \subseteq \mathcal{P}(\mathbb{E}) \times \mathcal{P}(\mathbb{E})$ to range over relations on observables.

Definition 9 *(Synchronous observations).* We say that two observations are synchronous under $\sqcap$ if, intuitively, the two following conditions hold:

1. every observable that can compose (under $\sqcap$) with another observable must occur simultaneously with one of its related observables; and
2. only an observable that does not compose (under $\sqcap$) with any other observable can happen before another observable, i.e., at a strictly lower time.

To formalize the conditions above, we use the independence relation $ind_\sqcap(X,Y) = \forall x \subseteq X. \forall y \subseteq Y. (x,y) \notin \sqcap$.

The *synchronous* composability relation on observations $\kappa^{sync}_\sqcap(E_1,E_2)$ is the smallest set such that, for all $O_1 \subseteq E_1$ and $O_2 \subseteq E_2$:

- if $(O_1,O_2) \in \sqcap \cup (\emptyset,\emptyset)$, then for all $(O_1',O_2') \in \mathcal{P}(E_1) \times \mathcal{P}(E_2)$ such that $ind_\sqcap(O_1',E_2)$ and $ind_\sqcap(E_1,O_2')$ and for all time stamps t, we have $((O_1 \cup O_1', t),(O_2 \cup O_2', t)) \in \kappa^{sync}_\sqcap(E_1,E_2)$;
- if $ind_\sqcap(O_1,E_2)$, then for all $O_2' \subseteq E_2$ and $t_1 < t_2$, we have $((O_1,t_1),(O_2,t_2)) \in \kappa^{sync}_\sqcap(E_1,E_2)$. Reciprocally, if $ind_\sqcap(E_1,O_2)$ then for all $O_1' \subseteq E_1$ and $t_2 < t_1$, we have $((O_1,t_1),(O_2,t_2)) \in \kappa^{sync}_\sqcap(E_1,E_2)$;

Example 12. Although the relation in Definition 9 is a binary relation on observations, we show in this example how to synchronize multiple events transitively. For instance, consider three components, $A = (\{a\}, L_A)$, $B = (\{b\}, L_B)$, and $C = (\{c\}, L_C)$. Let $\sqcap$ be the smallest symmetric relation with $\{(\{a\},\{b\}),(\{b\},\{c\})\} \subseteq \sqcap$. Then, $\kappa^{sync}_\sqcap$ enforces every observable in A and C to occur at the same time as an observable in B. Let $\Sigma = ([\kappa^{sync}_\sqcap], \cup)$ with $\cup$ defined in Example 3. Observe that, in general, $(A \times_\Sigma B) \times_\Sigma C \neq (A \times_\Sigma C) \times_\Sigma B$. On the left hand side, the product of A and B synchronizes every occurrence of event a with an occurrence of event b, which results in observables of the form $\{a,b\}$ only (no interleaving is allowed by $\kappa^{sync}_\sqcap$). Since b and c are also related events, the composition with C leads to the component with observables $\{a,b,c\}$. On the right hand side, A and C have independent observables and their composition allows for every interleaving. The product with B, however, synchronizes every occurrence of event a or c with an occurrence of event b, which results in interleaving of observables $\{a,b\}$ and $\{c,b\}$. Finally, observe that the component $(A \times_\Sigma B) \times_\Sigma C$ transitively synchronizes occurrences of event a with occurrences of event c through occurrences of event b. ■

The behavior of component $C_1 \times_{([\kappa^{sync}_\sqcap],\oplus)} C_2$ contains TESs obtained from the composition under $\oplus$ of every pair $\sigma_1 \in L_1$ and $\sigma_2 \in L_2$ of TESs that are related by the synchronous composability relation $[\kappa^{sync}_\sqcap]$ which, depending on $\sqcap$, excludes all event occurrences that do not synchronize.[6]

Definition 10 *(Mutual exclusion).* Let $\sqcap \subseteq \mathcal{P}(\mathbb{E})^2$ be a relation on observables. We define two observations to be mutually exclusive under the relation $\sqcap$ if no pair of observables in $\sqcap$ can be observed at the same time. The mutually exclu-

[6] If we let $\oplus$ be the element wise set union, define an event as a set of port assignments, and in the pair $([\kappa^{sync}_\sqcap], \oplus)$ let $\sqcap$ be true if and only if all common ports get the same value assigned, then this composition operator produces results similar to the composition operation in Reo [3].

sive composability relation $\kappa_{\sqcap}^{excl}$ on observations allows the composition of two observations (O_1, t_1) and (O_2, t_2), i.e., $((O_1, t_1), (O_2, t_2)) \in \kappa_{\sqcap}^{excl}(E_1, E_2)$, if and only if $t_1 = t_2 \implies \neg(O_1 \sqcap O_2)$.

Example 13. Following Example 10, we introduce an interaction signature that composes two robot-field subsystems while excluding the possibility for the robots to both observe the same location on their fields. We define $\sqcap = \{(\{(loc(R_1); l)\}, \{(loc(R_2); l)\}) \mid l \in [0; 20] \times [0; 20]\}$ as the set of pairs of observables containing, for both robots R_1 and R_2, an event that displays the same location as the other robot. Let $\Sigma = ([\kappa_{\sqcap}^{excl}], \cup)$ with $\cup$ defined in Example 3. Then, the product of the two subsystems, using the interaction signature Σ, excludes the possibility for the two robots to observe the same location at the same time. Strictly speaking, the exclusion imposed by the interaction signature Σ does not imply that the two robots can not *effectively* be on the same physical location. We show in Section 4 how, combined with hyper-properties, such interaction signature may imply a safety property. ■

Lemma 4. *If $\sqcap \subseteq \mathcal{P}(E_1) \times \mathcal{P}(E_2)$ is such that $(O, O) \in \sqcap$ for all non-empty $O \subseteq E_1 \cap E_2$, then $\bowtie(E_1, E_2) \subseteq [\kappa_{\sqcap}^{sync}](E_1, E_2)$ and $\between(E_1, E_2) \subseteq [\kappa_{\sqcap}^{excl}](E_1, E_2)$ where $\bowtie$ and $\between$ are defined in Example 4 and Example 5 respectively. When $E_1 \cap E_2 = \emptyset$, then $\bowtie(E_1, E_2) = \between(E_2, E_2)$.*

Proof. We show that $\bowtie(E_1, E_2) \subseteq \Phi_{\kappa_{sync}^{\sqcap}}(E_1, E_2)(\bowtie(E_1, E_2))$ and $\between(E_1, E_2) \subseteq \Phi_{\kappa_{excl}^{\sqcap}}(E_1, E_2)(\between(E_1, E_2))$. First, we observe that if $(\sigma, \tau) \in \bowtie(E_1, E_2)$, then $(\sigma, \tau)' \in \bowtie(E_1, E_2)$, as dropping the first observation(s) of σ and τ preserves the property imposed by $\bowtie$. Moreover, when $\sqcap$ is defined as above, $\kappa_{\sqcap}^{sync}$ is equivalent to κ^{sync} defined in Example 11, and for all $(\sigma, \tau) \in \bowtie(E_1, E_2)$, $(\sigma(0), \tau(0)) \in \kappa^{sync}(E_1, E_2)$. Since

$$\Phi_{\kappa^{sync}}(E_1, E_2)(\bowtie(E_1, E_2)) = \{(\sigma, \tau) \mid (\tau(0), \sigma(0)) \in \kappa^{sync}(E_1, E_2) \wedge (\sigma, \tau)' \in \bowtie(E_1, E_2)\}$$

we conclude that $\bowtie(E_1, E_2) \subseteq \Phi_{\kappa^{sync}}(E_1, E_2)(\bowtie(E_1, E_2))$. The proof is similar for $\between(E_1, E_2)$. □

The behavior of component $C_1 \times_{([\kappa_{\sqcap}^{excl}], \oplus)} C_2$ contains TESs resulting from the composition under $\oplus$ of every pair $\sigma_1 \in L_1$ and $\sigma_2 \in L_2$ of TESs that are related by the mutual exclusion composability relation $[\kappa_{\sqcap}^{excl}]$ which, depending on $\sqcap$, may exclude some simultaneous event occurrences.

The lifting of composability relations distributes across the intersection.[7]

Lemma 5. *For all composability relations κ_1, κ_2 and interfaces E_1, E_2:*

$$[\kappa_1 \cap \kappa_2](E_1, E_2) = [\kappa_1](E_1, E_2) \cap [\kappa_2](E_1, E_2)$$

Proof.

$$\begin{aligned}
[\kappa_1](E_1, E_2) \cap [\kappa_2](E_1, E_2) &= \bigcup\{\mathcal{R} \mid \mathcal{R} \subseteq \Phi_{\kappa_1}(E_1, E_2)(\mathcal{R})\} \cap \bigcup\{\mathcal{R} \mid \mathcal{R} \subseteq \Phi_{\kappa_2}(E_1, E_2)(\mathcal{R})\} \\
&= \bigcup\{\mathcal{R} \mid \mathcal{R} \subseteq \Phi_{\kappa_1}(E_1, E_2)(\mathcal{R}) \text{ and } \mathcal{R} \subseteq \Phi_{\kappa_2}(E_1, E_2)(\mathcal{R})\} \\
&= \bigcup\{\mathcal{R} \mid \mathcal{R} \subseteq \Phi_{\kappa_1}(E_1, E_2)(\mathcal{R}) \cap \Phi_{\kappa_2}(E_1, E_2)(\mathcal{R})\} \\
&= \bigcup\{\mathcal{R} \mid \mathcal{R} \subseteq \Phi_{\kappa_1 \cap \kappa_2}(E_1, E_2)(\mathcal{R})\} \\
&= [\kappa_1 \cap \kappa_2](E_1, E_2)\}
\end{aligned}$$

since

$$\begin{aligned}
&\Phi_{\kappa_1}(E_1, E_2)(\mathcal{R}) \cap \Phi_{\kappa_2}(E_1, E_2)(\mathcal{R}) \\
&\quad = \{(\tau_1, \tau_2) \mid (\tau_1(0), \tau_2(0)) \in \kappa_1(E_1, E_2) \wedge (\tau_1(0), \tau_2(0)) \in \kappa_2(E_1, E_2) \wedge (\tau_1, \tau_2)' \in \mathcal{R})\} \\
&\quad = \{(\tau_1, \tau_2) \mid (\tau_1(0), \tau_2(0)) \in \kappa_1(E_1, E_2) \cap \kappa_2(E_1, E_2) \wedge (\tau_1, \tau_2)' \in \mathcal{R})\} \\
&\quad = \Phi_{\kappa_1 \cap \kappa_2}(E_1, E_2)(\mathcal{R}) \quad \square
\end{aligned}$$

Similarly, we give a mechanism to lift a composition function on observables to a composition function on TESs. Such lifting operation interleaves observations with different time stamps, and composes observations that occur at the same time.

[7] The lifting does not distribute across the union, however.

Definition 11 *(Lifting - composition function).* Let $+ : \mathcal{P}(\mathbb{E}) \times \mathcal{P}(\mathbb{E}) \to \mathcal{P}(\mathbb{E})$ be a composition function on observables. The *lifting* of $+$ to TESs is $[+] : TES(\mathbb{E}) \times TES(\mathbb{E}) \to TES(\mathbb{E})$ such that, for $\sigma_i \in TES(\mathbb{E})$ where $\sigma_i(0) = (O_i, t_i)$ with $i \in \{1, 2\}$:

$$\sigma_1[+]\sigma_2 = \begin{cases} \langle \sigma_1(0) \rangle \cdot (\sigma_1'[+]\sigma_2) & \text{if } t_1 < t_2 \\ \langle \sigma_2(0) \rangle \cdot (\sigma_1[+]\sigma_2') & \text{if } t_2 < t_1 \\ \langle (O_1 + O_2, t_1) \rangle \cdot (\sigma_1'[+]\sigma_2') & \text{otherwise} \end{cases}$$

Definition 11 composes observations only if their time stamp is the same. Alternative definitions might consider time intervals instead of exact times.

Remark 1. The last clause of Definition 11 considers the case where two observations occur at the same time. Recall that the time of an observation, as introduced earlier, is an abstraction that requires every event of the observation to occur after the events of the previous observation, and before the events of the next observation. Moreover, the time of two related observations, during composition, may be constrained by the interaction signature of the composition. For instance, the synchronous composability relation in Example 4 requires related observations to occur at the same time. Given those two facts, the likelihood that two observations have the same time is non zero.

Example 14 *(Intersection).* For any two components $C_1 = (E_1, L_1)$ and $C_2 = (E_2, L_2)$, we define the intersection $C_1 \cap C_2$ to be the component $C_1 \times_{([\kappa_\sqcap^{sync}],[\cap])} C_2 = (E_1 \cup E_2, L)$ where $\sqcap \subseteq E_1 \times E_2$ is such that $(O, O) \in \sqcap$ for all non-empty $O \subseteq E_1 \cup E_2$. ■

Example 15 *(Join).* For any two components $C_1 = (E_1, L_1)$ and $C_2 = (E_2, L_2)$, we define the join $C_1 \bowtie C_2$ to be the component $C_1 \times_{([\kappa_\sqcap^{sync}],[\cup])} C_2 = (E_1 \cup E_2, L)$ where $\sqcap \subseteq E_1 \times E_2$ is such that $(O, O) \in \sqcap$ for all non-empty $O \subseteq E_1 \cap E_2$. Note that the join of two components contains more behavior than the intersection of those two components: independent events (i.e., events not in $E_1 \cap E_2$) may occur freely in any observation. If $E_1 = E_2$, however, then $C_1 \bowtie C_2 = C_1 \cap C_2$. ■

Lemma 6. *Let κ_1 and κ_2 be two composability relations and $\times_{([\kappa_1 \cap \kappa_2],\oplus)}$ be a product on components. Then,*

$$C_1 \times_{([\kappa_1 \cap \kappa_2],\oplus)} C_2 = C_1 \times_{([\kappa_1] \cap [\kappa_2],\oplus)} C_2 = (C_1 \times_{([\kappa_1],\oplus)} C_2) \cap (C_1 \times_{([\kappa_2],\oplus)} C_2)$$

Proof. Let $C_1 \times_{[\kappa_1 \cap \kappa_2],\oplus} C_2 = (E, L)$ and $(C_1 \times_{([\kappa_1],\oplus)} C_2) \cap (C_1 \times_{([\kappa_2],\oplus)} C_2) = (E', L')$. We have $E = E_1 \cup E_2 = E'$. We show $L = L'$.

$$\begin{aligned} L &= \{\sigma_1 \oplus \sigma_2 \mid \sigma_1 \in L_1, \sigma_2 \in L_2, (\sigma_1, \sigma_2) \in [\kappa_1 \cap \kappa_2](E_1, E_2)\} \\ &= \{\sigma_1 \oplus \sigma_2 \mid \sigma_1 \in L_1, \sigma_2 \in L_2, (\sigma_1, \sigma_2) \in [\kappa_1](E_1, E_2) \cap [\kappa_2](E_1, E_2)\} \\ &= L' \quad \square \end{aligned}$$

Example 16. Composability relations as defined in Definition 9 and Definition 10 can be combined to form new relations, and therefore new products. The behavior of component $C_1 \times_{([\kappa_\sqcap^{sync} \cap \kappa_\sqcap^{excl}]),\oplus)} C_2$ contains all TESs that are in the behavior of both $C_1 \times_{([\kappa_\sqcap^{sync}],\oplus)} C_2$ and $C_1 \times_{([\kappa_\sqcap^{excl}],\oplus)} C_2$, which excludes observations containing an occurrence of at least one event related by $\sqcap$. ■

To show the expressiveness of our model, we consider its application to the Reo coordination language [3] in a series of example (Examples 17, 18). Reo is a coordination language for components, and makes use of shared ports between components to synchronize their observables. If two components share a port, then they "agree" on the value of the data that flow through that port. A component has a set of ports as interface, and defines a relation over its port values over time. Composition of two components is taking the conjunction of their relations.

Example 17 *(Reo components).* Let P be the universal set of port names in Reo, and $V(p)$ the set of values over port $p \in P$. Consider the set of events $\mathbb{E} = \{(p, v) \mid p \in P,\ v \in V(p)\}$ that contains all pairs of a port name and a value, and $E_a \subseteq \mathbb{E}$ be the set of all events for port a only, i.e., $E_a = \{(a, v) \mid v \in V(a)\}$. A port $a \in P$ corresponds to the component $C_a = (E_a, TES(E_a))$.

For any ports a, b and function $f : V(a) \to V(b)$, we introduce a composability relation on observables $\sqcap_{(a,b,f)} \subseteq \mathcal{P}(E_a) \times \mathcal{P}(E_b)$ such that, for all $(O_1, O_2) \in \sqcap_{(a,b,f)}$, there exists $v \in V(a)$ with $(a, v) \in O_1$ if and only if $(b, f(v)) \in O_2$. Intuitively, such composability relation relate two observables O_1 and O_2 such that the value of b in O_2 coincides with the image of the value of a in O_1 under f. We omit f when f is the identity, and write $\sqcap_{(a,b)}$. We use $\cup$ as composition function on observables and write $\times_R$ as a shorthand for $\times_{(R,[\cup])}$.

The composability relation $\kappa^{intl}_{\sqcap}$ interleaves observations while restricting every occurrence of observable O_1 to be related to a later observable O_2, i.e., $((O_1, t_1), (O_2, t_2)) \in \kappa^{intl}_{\sqcap}$ if and only if

$$t_2 < t_1 \vee (t_1 < t_2 \wedge (O_1, O_2) \in \sqcap)$$

We define the alternating component: $M = (E_m, L_m)$ where $E_m = \{(m, 0), (m, 1)\}$ and $\sigma \in L_m \subseteq TES(E_m)$ if and only if, for all $i \in \mathbb{N}$, $\sigma(2i) = (\{(m, 0)\}, t_i)$ and $\sigma(2i + 1) = (\{(m, 1)\}, t_{i+1})$, which consists of a stream of alternating bits. Then, fixing the function f_0 such that $f_0(v) = 0$ for all $v \in V(a)$, the product $P_a \times_{\kappa^{sync}_{\sqcap(a,m,f_0)}} M$ represents the component that synchronizes all values at port a with the value 0 at port m. Reciprocally, fixing the function f_1 such that $f_1(v) = 1$ for all $v \in V(b)$, the product $P_b \times_{\kappa^{sync}_{\sqcap(b,m,f_1)}} M$ represents the component that synchronizes all values at port b with the value 1 at port m.

We define the following Reo components:

$$\begin{aligned}
Sync(a, b) &= P_a \times_{[\kappa^{sync}_{\sqcap(a,b)}]} P_b \\
Syncdrain(a, b) &= P_a \times_{[\kappa^{sync}_{\sqcap}]} P_b \text{ with } \sqcap = \mathcal{P}(E_a) \times \mathcal{P}(E_b) \\
Fifo(a, b, M) &= (P_a \times_{[\kappa^{sync}_{\sqcap(a,m,f_0)}]} M) \times_{[\kappa^{intl}_{\sqcap(a,b)}]\cup[\kappa^{sync}_{\sqcap(m,m)}]} (P_b \times_{[\kappa^{sync}_{\sqcap(b,m,f_1)}]} M) \\
Merger(a, b, c) &= (P_a \times_{[\kappa^{excl}_{\sqcap(a,b)}]} P_b) \times_{[\kappa^{sync}_{\sqcap(a,c)\cup\sqcap(b,c)}]} P_c
\end{aligned}$$

The $Sync(a, b)$ component is such that the data observed at port a and b are equal and synchronous, i.e., occurs at the same time. The $Syncdrain(a, b)$ component ensures that both the data of a and b are observed at the same time, but does not restrict their data to be equal. The component $Fifo(a, b, M)$ synchronizes the observation of a data at a with the change of the memory state M, and then outputs the same data at b. As defined here, the $Fifo(a, b, M)$ component is infinitely productive, i.e., always eventually has an input at a and an output at b. The $Merger(a, b, c)$ component either synchronizes a with c or b with c but never all ports together.

A strength of Reo is its compositional nature: protocols are built out of primitives. We use the join operation defined in Example 15 (see Lemma 8 for the proof of associativity and commutativity of $\bowtie$) to define two Reo components:

$$\begin{aligned}
Alternator(a, b, c) &= Sync(a, c_1) \bowtie Fifo(x, c_2) \bowtie Syncdrain(a, b) \bowtie Sync(b, x) \bowtie Merger(c_1, c_2, c) \\
Fifo_2(a, b) &= Fifo(a, x, M_1) \bowtie Fifo(x, b, M_2) \quad \blacksquare
\end{aligned}$$

In order to state some sufficient conditions for κ so that its lifted product satisfies the condition for associativity, we introduce the unique *enumeration* of a triple of TESs. Intuitively, the enumeration is a stream that increments the index of each TES in order. Observe, from Definition 12, that for any triple $(\sigma_1, \sigma_2, \sigma_3)$, there exists a unique such enumeration. Intuitively, the enumeration τ keeps a counter for each TES in the triple such that, for all steps, the counters increment with the smallest value that progresses time.

Definition 12 *(Enumeration).* Let $(\sigma_1, \sigma_2, \sigma_3) \in TES(E_1) \times TES(E_2) \times TES(E_3)$, an enumeration $\tau : \mathbb{N} \to (\mathbb{N} \times \mathbb{N} \times \mathbb{N})$ is inductively defined by $\tau(0) = (0, 0, 0)$ and, given $\tau(n) = (i_1, i_2, i_3)$ and $t = \min(\{\mathrm{pr}_2(\sigma_1)(i_1), \mathrm{pr}_2(\sigma_2)(i_2), \mathrm{pr}_2(\sigma_3)(i_3)\})$,

$$\tau(n + 1) = (i'_1, i'_2, i'_3) \text{ such that } i'_k = \min(i \mid i_k \leq i \text{ and } t \neq \mathrm{pr}_2(\sigma_k)(i) \text{ for } k \in \{1, 2, 3\}$$

Co-inductively construction of interaction signatures make proving algebraic results of Lemma 1 easier. Lemma 7 gives sufficient conditions to lift, by co-induction, properties of the underlying relation and composition function to meet the conditions of Lemma 1.

Lemma 7. *Let $+$ be a composition function on observables and let κ be a composability relation on observations. Then,*

- $\times_{([\kappa],[+])}$ *is* commutative *if κ is symmetric and $+$ is commutative;*
- $\times_{([\kappa],[+])}$ *is* associative *if $+$ is associative and, for all E_1, E_2, E_3 and for all $(\sigma_1, \sigma_2, \sigma_3) \in TES(E_1) \times TES(E_2) \times TES(E_3)$, let τ be an enumeration of $(\sigma_1, \sigma_2, \sigma_3)$, then, for all $n \in \mathbb{N}$, letting $\tau(n) = (i, j, k)$,*

  $$(\sigma_1(i), \sigma_2(j)) \in \kappa(E_1, E_2) \wedge ((\sigma_1^{(i)}[+]\sigma_2^{(j)})(0), \sigma_3(k)) \in \kappa(E_1 \cup E_2, E_3)$$

  if and only if, for all $n \in \mathbb{N}$, letting $\tau(n) = (i, j, k)$,

  $$(\sigma_2(j), \sigma_3(k)) \in \kappa(E_2, E_3) \wedge (\sigma_1(i), (\sigma_2^{(j)}[+]\sigma_3^{(k)})(0)) \in \kappa(E_1, E_2 \cup E_3)$$

- $\times_{([\kappa],[+])}$ *is* idempotent *if $+$ is idempotent and, for all $E \subseteq \mathbb{E}$ we have $((O_1, t_1), (O_2, t_2)) \in \kappa(E, E) \implies (O_1, t_1) = (O_2, t_2)$.*

Proof. Proof in Appendix A. □

The result of Lemma 7 can be expressed in terms of conditions on composability relations on observables. In Property 1, we introduce some sufficient conditions for $\times_{([\kappa_\sqcap^{sync}],[\cup])}$ to be associative (i.e., to satisfy conditions in Lemma 7).

Property 1. We list a series of properties on a composability relation on observable $\sqcap$ where, for $O_1, O_2, O_3 \subseteq \mathbb{E}$:

1. if $ind_\sqcap(O_1, O_2)$ and $ind_\sqcap(O_1, O_3)$ then $ind_\sqcap(O_1, O_2 \cup O_3)$; and if $ind_\sqcap(O_2, O_3)$ and $ind_\sqcap(O_1, O_3)$ then $ind_\sqcap(O_1 \cup O_2, O_3)$
2. if $(O_1, O_2) \in \sqcap$ and $(O_2, O_3) \in \sqcap$, then $(O_1, O_3) \in \sqcap$;
3. if $(O_1, O_2) \in \sqcap$ and $(O_1, O_3) \in \sqcap$, then $(O_2, O_3) \in \sqcap$ and $(O_3, O_2) \in \sqcap$;
4. if $(O_1, O_3) \in \sqcap$ and $(O_2, O_3) \in \sqcap$, then $(O_1, O_2) \in \sqcap$ and $(O_2, O_1) \in \sqcap$;
5. if $(O_1, O_2) \in \sqcap$ and $(O'_1, O'_2) \in \sqcap$, then $(O_1 \cup O'_1, O_2 \cup O'_2) \in \sqcap$.

Lemma 8. *Let $\sqcap$ be a composability relation on observables and $\kappa_\sqcap^{sync}$ as defined in Definition 9. Then:*

- *if $\sqcap$ is symmetric, then the product $\times_{([\kappa_\sqcap^{sync}],[\cup])}$ is commutative;*
- *if $\sqcap$ satisfies Property 1, then the product $\times_{([\kappa_\sqcap^{sync}],[\cup])}$ is associative;*
- *if $\sqcap$ is co-reflexive, i.e., $(O_1, O_2) \in \sqcap$ implies $O_1 = O_2$, then the product $\times_{([\kappa_\sqcap^{sync}],[\cup])}$ is idempotent.*

Proof. Proof in Appendix A. □

Example 18 *(Properties of join).* Let $\sqcap = \{(O, O) \mid O \subseteq \mathbb{E}\}$ with $\mathbb{E}$ the universal set of events, then $\sqcap$ is co-reflexive, symmetric, and satisfies each item of Property 1. As a consequence, the *join* product $\bowtie$ defined in Example 15 is commutative, associative, and idempotent. Note that the join product is the product operator for Reo components.

We give in Lemma 9 some conditions for two products to distribute, and in Lemma 10 some conditions to extend the underlying relation on observables for a synchronous composability relation.

Lemma 9. *Let C_1, C_2, and C_3 be three components, and let κ_1 and κ_2 be two composability relations on observables such that for all $\sigma_1, \sigma_2, \sigma_3 \in L_1 \times L_2 \times L_3$:*

- *$(\sigma_1, \sigma_2[\cup]\sigma_3) \in [\kappa_1]$ if and only if $(\sigma_1, \sigma_2) \in [\kappa_1]$ and $(\sigma_1, \sigma_3) \in [\kappa_1]$, and*
- *for all $\tau_1 \in L_1$, $(\tau_1[\cup]\sigma_2, \sigma_1[\cup]\sigma_3) \in [\kappa_2]$ if and only if $(\sigma_2, \sigma_3) \in [\kappa_2]$ and $\sigma_1 = \tau_1$.*

Then,

$$C_1 \times_{[\kappa_1]} (C_2 \times_{[\kappa_2]} C_3) = (C_1 \times_{[\kappa_1]} C_2) \times_{[\kappa_2]} (C_1 \times_{[\kappa_1]} C_3)$$

Proof. Let L be the behavior of component $(C_1 \times_{[\kappa_1]} C_2) \times_{[\kappa_2]} (C_1 \times_{[\kappa_1]} C_3)$, L' be the behavior of $C_1 \times_{[\kappa_1]} (C_2 \times_{[\kappa_2]} C_3)$, L_{12} be the behavior of $(C_1 \times_{[\kappa_1]} C_2)$ and L_{13} be the behavior of $(C_1 \times_{[\kappa_1]} C_3)$. Then,

$$\begin{aligned}
L =& \{\sigma_1[\cup](\sigma_2[\cup]\sigma_3) \mid \sigma_1 \in L_1,\ \sigma_2 \in L_2,\ \sigma_3 \in L_3,\ (\sigma_1, \sigma_2[\cup]\sigma_3) \in [\kappa_1],\ (\sigma_2, \sigma_3) \in [\kappa_2]\} \\
=& \{\sigma_1[\cup](\sigma_2[\cup]\sigma_3) \mid \sigma_1 \in L_1,\ \sigma_2 \in L_2,\ \sigma_3 \in L_3,\ (\sigma_1, \sigma_2) \in [\kappa_1],\ (\sigma_1, \sigma_3) \in [\kappa_1],\ (\sigma_2, \sigma_3) \in [\kappa_2]\} \\
=& \{\sigma[\cup]\tau \mid \sigma \in L_{12},\ \tau \in L_{13},\ (\sigma, \tau) \in [\kappa_2]\} \\
=& L' \quad \square
\end{aligned}$$

Lemma 10. *Let $C_1 = (E_1, L_1)$ and $C_2 = (E_2, L_2)$ be two components. Let $\kappa_\sqcap^{sync}$ be a composability relation on observables with $\sqcap \subseteq \mathcal{P}(E_1) \times \mathcal{P}(E_2)$. Then, for any $\sqcap'$ with $\sqcap' \cap (\mathcal{P}(E_1) \times \mathcal{P}(E_2)) = \emptyset$, then:*

$$C_1 \times_{[\kappa_\sqcap^{sync}]} C_2 = C_1 \times_{[\kappa_{\sqcap\cup\sqcap'}^{sync}]} C_2$$

Proof. For any observations $((O_1, t_1), (O_2, t_2)) \in \kappa_\sqcap^{sync}$, we have that $((O_1, t_1), (O_2, t_2)) \in \kappa_{\sqcap\cup\sqcap'}^{sync}$ since $(O_1, O_2) \in \sqcap$ implies that $(O_1, O_2) \in \sqcap \cup \sqcap'$. Conversely, if $(O_1, O_2) \in \sqcap \cup \sqcap'$ and $\sqcap' \cap \mathcal{P}(E_1) \times \mathcal{P}(E_2) = \emptyset$, then $(O_1, O_2) \in \sqcap$. Thus, for any (σ_1, σ_2), $(\sigma_1, \sigma_2) \in [\kappa_\sqcap^{sync}]$ if and only if $(\sigma_1, \sigma_2) \in [\kappa_{\sqcap\cup\sqcap'}^{sync}]$. □

3.5. Properties of TESs

We distinguish two kinds of properties of TESs: properties that we call *trace properties*, and properties on sets of TESs that we call *behavior properties*, which correspond to hyper-properties in [6]. The generality of our model permits to interchangeably construct a component from a property and extract a property from a component. As illustrated in Example 21, when composed with a set of interacting components, a component property constrains the components to only expose desired behavior (i.e., behavior in the property). In Section 4, we provide more intuition for the practical relevance of these properties.

Definition 13. A trace property P is a subset $P \subseteq TES(E)$ for some set of events E. A component $C = (E, L)$ satisfies a property P, if $L \subseteq P$, which we denote as $C \models P$.

Example 19. We distinguish the usual *safety* and *liveness* properties [9,6], and recall that every trace property can be written as the intersection of a safety and a liveness property. Let X be an arbitrary set, and P be a subset of $\mathbb{N} \to X$. Intuitively, P is safe if every *bad* stream not in P has a finite prefix every completion of which is bad, hence not in P. A property P is a liveness property if every finite sequence in X^* can be completed to yield an infinite sequence in P, where X^* is the set of all finite sequences of elements in X. For instance, the property of terminating behavior for a component with interface E is a liveness property, defined as:

$$P_{finite}(E) = \{\sigma \in TES(E) \mid \exists n \in \mathbb{N}. \forall i > n.\ \mathrm{pr}_1(\sigma)(i) = \emptyset\}$$

$P_{finite}(E)$ says that, for every finite prefix of any stream in $TES(E)$, there exists a completion of that prefix with an infinite sequence of silent observations $\emptyset$ in $P_{finite}(E)$. ■

Example 20. A trace property is similar to a component, since it describes a set of TESs, except that it is *a priori* not restricted to any interface.[8] A trace property P can then be turned into a component, by constructing the smallest interface E_P such that, for all $\sigma \in P$, and $i \in \mathbb{N}$, $\mathrm{pr}_1(\sigma)(i) \subseteq E_P$. The component $C_P = (E_P, P)$ is then the componentized-version of property P. ■

Lemma 11. *Given a property P over E, its componentized-version C_P (see Example 20) and a component $C = (E, L)$, then $C \models P$ if and only if $C \cap C_P = C$.*

Proof. We recall the definition of the intersection in Example 14. For any two components $C_1 = (E_1, L_1)$ and $C_2 = (E_2, L_2)$, the intersection $C_1 \cap C_2$ is the component $C_1 \times_{([\kappa_\sqcap^{sync}],[\cap])} C_2 = (E_1 \cup E_2, L)$ where $\sqcap \subseteq E_1 \times E_2$ is such that $(O, O) \in \sqcap$ for all non-empty $O \subseteq E_1 \cup E_2$. Given that $\cap$ satisfies the condition for using Lemma 8, the product $\cap$ is idempotent. Let $C \cap C_P = (E, L')$. If $(\sigma, \tau) \in L'$ then $\sigma = \tau$. Thus, $L' \subseteq L \cap L_P$.

Alternatively, let $\sigma \in L \cap L_P$. We observe that at any point $n \in \mathbb{N}$, we have $(\sigma(n), \sigma(n)) \in \kappa_\sqcap^{sync}(E, E)$, since, given $\sigma(n) = (O_n, t_n)$ and the assumption on $\sqcap$, we have $(O_n, O_n) \in \sqcap$ or $O_n = \emptyset$. Therefore, $(\sigma, \sigma) \in [\kappa_\sqcap^{sync}]$.

We conclude that $L \cap L_P = L'$. □

Example 21. We use the term *coordination property* to refer to a property used in order to coordinate behaviors. Given a set of n components $C_i = (E_i, L_i)$, $i \in \{1, ..., n\}$, a coordination property *Coord* for the composed components is a property over events $E = E_1 \cup ... \cup E_n$, i.e., $Coord \subseteq TES(E)$.

Consider the synchronous interaction, as introduced in Example 15, of the n components and let $C = C_1 \bowtie C_2 \bowtie ... \bowtie C_n$ be their synchronous product. Typically, a coordination property will not necessarily be satisfied by the composite component C, but some of the behavior of C is contained in the coordination property. The coordination problem is to find (e.g., synthesize) an orchestrator component $Orch = (E_O, L_O)$ such that $C \bowtie Orch \models Coord$. The orchestrator restricts the component C to exhibit only the subset of its behavior that satisfies the coordination property. In other words, in their composition, *Orch* coordinates C to satisfy *Coord*. As shown in Example 20, since *Coord* ranges over the same set E that is the interface of component $C_1 \bowtie C_2 \bowtie ... \bowtie C_n$, a coordination property can be turned into an orchestrator by building its corresponding component. The coordination problem can be made even more general by changing the composability relations or the composition functions used in the construction of C. ■

Trace properties are not sufficient to fully capture the scope of interesting properties of components of cyber-physical systems. Some of their limitations are highlighted in Section 4. To address this issue, we introduce *behavior properties*, which are strictly more expressive than trace properties, and give two illustrative examples.

[8] In our formalism, a property is a set of TESs $L \subseteq TES(E)$ for some $E \subseteq \mathbb{E}$. Two properties P and L are equal if they contain identical TESs, and equality is not subject to the interface over which properties are defined.

Definition 14. A behavior property ϕ over a set of events E is a hyper-property $\phi \subseteq \mathcal{P}(TES(E))$. A component $C = (E, L)$ satisfies a hyper-property ϕ if $L \in \phi$, which we denote as $C \models \phi$.

Example 22. A component $C = (E, L)$ can be oblivious to time. Any sequence of time-stamps for an acceptable sequence of observables is acceptable in the behavior of such a component. This "obliviousness to time" property is not a trace property, but a hyper-property, defined as:

$$\phi_{shift}(E) := \{Q \subseteq TES(E) \mid \forall \sigma \in Q.\forall t \in OS(\mathbb{R}_+).\exists \tau \in Q.\mathsf{pr}_1(\sigma) = \mathsf{pr}_1(\tau) \wedge \mathsf{pr}_2(\tau) = t\}$$

Intuitively, if $C \models \phi_{shift}(E)$, then C is independent of time. ■

Example 23. We use $\phi_{insert}(X, E)$ to denote the hyper-property that allows for arbitrary insertion of observations in $X \subseteq \mathcal{P}(E)$ into every TES at any point in time, i.e., the set defined as:

$$\{Q \subseteq TES(E) \mid \forall \sigma \in Q.\forall i \in \mathbb{N}.\exists \tau \in Q.\exists x \in X.(\forall\ j < i.\quad \sigma(j) = \tau(j)) \wedge (\exists\ t \in \mathbb{R}_+.\quad \tau(i) = (x, t)) \wedge (\forall\ j \geq i.\quad \tau(j+1) = \sigma(j))\ \}$$

Intuitively, elements of $\phi_{insert}(X, E)$ are closed under insertion of an observation $x \in X$ at an arbitrary time. ■

4. Application

This section is inspired by the work on soft-agents [10,11], and elaborates on the more intuitive version that we presented in Section 2. We show in Sections 4.1 and 4.2 some expressions that represent interactive cyber-physical systems, and in Section 4.3 we formulate some trace and behavior properties of those systems. Through these examples, we show how we use component-based descriptions to model a simple scenario of a robot roaming around in a field while taking energy from its battery. We structurally separate the battery, the robot, and the field as independent components, and we explicitly model their interaction in a specific composed system.

4.1. Description of components

We give, in order, a description for a robot, a battery, and a field component. Each component reflects a local and concise view of the physical and cyber aspects of the system.

Robot. A robot component, with identifier R and reading sensors *loc* and *bat*, is a tuple $R = (E_R, L_R)$ with:

$$\begin{aligned} E_R &= \{(read(loc, R); l), (read(bat, R); b), (move(R); (d, \alpha)), (charge(R); \mathsf{ON}) \mid \\ &\qquad l \in [0, 20]^2,\ b \in \mathbb{R}_+,\ d \in \{\mathsf{N}, \mathsf{E}, \mathsf{W}, \mathsf{S}\},\ \alpha \in \mathbb{R}_+\} \\ L_R &= \{\sigma \in TES(E_R) \mid \forall i \in \mathbb{N}.\exists e \in E_R.\ \mathsf{pr}_1(\sigma)(i) = \{e\}\} \end{aligned}$$

where the read of the position, the read of the battery, the move, and the charge events contain respectively the position of the robot as a pair of coordinates $l \in [0, 20]^2$ grid; the remaining battery power b (in Wh); the move direction as pair of a cardinal direction d and a positive number α for a demand of energy (in W); and the charge status as ON. Note that the set of TESs L_R allows for arbitrary increasing and non-Zeno sequences of timestamp.

Battery. A battery component, with identifier B and capacity C (in Wh), is a tuple $B(C) = (E_B, L_B)$ with:

$$\begin{aligned} E_B &= \{(read(B); l), (discharge(B); \eta_d), (charge(B); \eta_c) \mid 0 \leq l \leq C,\ \eta_c, \eta_d : \mathbb{R}_+ \to \mathbb{R}_+\} \\ L_B &= \{\sigma \in TES(E_B) \mid \forall i \in \mathbb{N}.\exists e \in E_B.\ \mathsf{pr}_1(\sigma)(i) = \{e\} \wedge P_B(\sigma)\} \end{aligned}$$

where the read, the charge, and the discharge events respectively contain the current charge status l (in Wh), the discharge rate η_d, and the charge rate η_c. The discharge and charge rates are coefficients that depend on the internal constitution of the battery, e.g., its current and voltage, and influence how the battery supplies energy to its user. The integration of η_d (or η_c) on a time interval gives the power delivered (or received) by the battery in Wh. The predicate $P_B(\sigma)$ guarantees that every behavior σ of the battery satisfies the physical constraints for its acceptability. An example for the structural constraint P_B is that every *read* event instance returns the battery level as a function of the occurrence of prior *discharge* and *charge* events. We introduce the *lev* function, that takes a sequence of observations $s \in (\mathcal{P}(\mathbb{E}) \times \mathbb{R}_+)^*$ of size $i > 1$ and returns the cumulative energy spent: $lev(s) = lev(\langle s(0), s(1)\rangle) + lev(\langle s(1)...s(i)\rangle)$, where

$$lev(\langle (O_1, t_1), (O_2, t_2)\rangle) = \begin{cases} \int_{t_1}^{t_2} (\eta_d(t) + \eta_I(t))dt & if\ (discharge(B); \eta_d) \in O_1 \\ \int_{t_1}^{t_2} (\eta_I(t) - \eta_c(t))dt & if\ (charge(B); \eta_c) \in O_1 \\ \int_{t_1}^{t_2} \eta_I(t)dt & \text{otherwise} \end{cases}$$

and η_I is an internal discharge rate. The constraint P_B is defined such that all $(read(B); l)$ events in E_B return the current battery level of the robot, in accordance with the lev function, i.e., for all $\sigma \in L_B$:

$$\forall i \in \mathbb{N}.(read(B); l) \in \mathsf{pr}_1(\sigma)(i) \implies l = \min(C, \max(C - lev(\langle \sigma(0), ..., \sigma(i)\rangle), 0))$$

where C is the maximum capacity of the battery. The property P_B assumes that initially at $t = 0$ the battery is at its maximum charge C, that the battery level decreases after each discharge event, increases after each charge event, proportionally to the discharge and the charge rates. Moreover, a discharge below 0 is physically forbidden. Observe that different alternatives for the predicate P_B account for different models of batteries. Moreover, our model allows for specifications where the discharge factor depends on external parameters (temperature, discharge level, etc), adding a non-linear aspect to the model.

Field. A field component $F(l_0)$ contains a single object that we identify as I initially at location l_0, has a fixed size of $[0, 20]^2$, and contains a charging station at location $(5; 5)$. A field component is a tuple (E_F, L_F) with:

$$E_F = \{(loc(I); p), (move(I); (d, F_t)) \mid p \in [0, 20]^2,\ d \in \{\mathsf{N}, \mathsf{S}, \mathsf{E}, \mathsf{W}\},\ F_t \in \mathbb{R}_+\}$$

$$L_F = \{\sigma \in TES(E_F) \mid \forall i \in \mathbb{N}.\exists e \in E_F.\, \mathsf{pr}_1(\sigma)(i) = \{e\} \wedge P_F(\sigma)\}$$

where the *loc* and the *move* events respectively contain the position of object I and the pair of a direction d of the move of object I and a force F_t of traction applied by object I. A field has an internal friction factor μ whose value depends on the position on the field. With a friction value of 0, the object will have no traction and thus will stay put in place instead of moving on the field (e.g., failure to move on a layer of ice). With a friction of 1, the move event will displace the object proportionally to the force of the move (e.g., a move on a layer of concrete). A friction factor between 0 and 1 captures other scenarios between those two extremes (e.g., a move on a layer of grass). The predicate $P_F(\sigma)$ guarantees that every behavior σ satisfies the physics of the field component. P_F models the case where the object I is initially at position l_0 and every move event changes continuously the location of the object on the field according to the direction d, the force of traction F_t, and the friction μ. A move event has no effect if it occurs while the position of I is on the boundary of the field: this scenario simulates the case of a fenced field, where moving against the fence has the same observable as not moving.

The internal constraints of the field are such that the *move* observation triggers an internal displacement of object I proportional to the force that the object has applied. We write $\Delta d(t, t_0, (x_0, y_0))$ to denote the displacement from a time t_0 where the object is at rest at position (x_0, y_0), to a time t, defined as:

$$\begin{aligned} m\vec{a} &= \vec{F}_t(x_0, y_0) \\ ma &= F_t(x_0, y_0) \\ v(t, t_0) &= \left(\frac{F_t(x_0, y_0)}{m}\right)(t - t_0) \\ \Delta d(t, t_0, (x_0, y_0)) &= \frac{1}{2}\left(\frac{F_t(x_0, y_0)}{m}\right)(t - t_0)^2 \end{aligned} \tag{1}$$

where $||\vec{F}_t|| \leq \frac{1}{4}\,\mu(x_0, y_0)mg$, e.g., the traction force on a wheel (supporting one fourth of the weight of the object) is less than the maximal friction force, with μ the friction coefficient, m the mass of the object; and F_t is the constant traction force of the object. Observe that we chose to make the friction coefficient dependent on the initial position x_0 of the object before the move. This choice reflects the simplifying assumption that the friction will not substantially change during the movement. Alternatively, one can imagine a different structure for the field component to support variable friction during a move in P_F.

An example for the constraint P_F reflects the constraint that for each sequence of observations, the output value of a *read* event corresponds to the current position of the robot given its previous moves. We will use a function called *dis* to determine the cumulative displacement of the robot after a sequence of observations. Let $s \in (\mathcal{P}(\mathbb{E}) \times \mathbb{R}_+)^*$ be a finite sequence of observations of size $i \geq 1$. The displacement of the object I, at position (x_0, y_0), after a sequence of events s is given by $dis(\langle (O_0, t_0)\rangle, (x_0, y_0)) = (x_0, y_0)$ and $dis(s, (x_0, y_0)) = dis(\langle s(0), ..., s(i-1)\rangle, (x', y'))$, where for $s(0) = (O_0, t_0)$ and $s(1) = (O_1, t_1)$:

$$(x', y') = \begin{cases} (x_0, y_0 + \Delta d(t_1, t_0, (x_0, y_0))) & if\ (move(I); (\mathsf{N}, F_t)) \in O_0 \\ (x_0, y_0 - \Delta d(t_1, t_0, (x_0, y_0))) & if\ (move(I); (\mathsf{S}, F_t)) \in O_0 \\ (x_0 + \Delta d(t_1, t_0, (x_0, y_0)), y_0) & if\ (move(I); (\mathsf{E}, F_t)) \in O_0 \\ (x_0 - \Delta d(t_1, t_0, (x_0, y_0)), y_0) & if\ (move(I); (\mathsf{W}, F_t)) \in O_0 \\ (x_0, y_0) & \text{otherwise} \end{cases}$$

with $\Delta d(t, t_0, (x_0, y_0))$ defined in Equation (1). P_F is defined to accept all TESs such that every *read* event returns the current position of the robot on the field, according to its displacement over time. Given $\sigma \in TES(E_F)$, $P_F(\sigma)$ is true if and only if

$$\forall i \in \mathbb{N}. \quad (loc(I); p) \in \mathrm{pr}_1(\sigma)(i) \implies p = |dis(\langle \sigma(0) .... \sigma(i) \rangle, l_0)|_{[-20,20]}$$

with $|(x, y)|_{[-20,20]} = (\min(\max(x, -20), 20), \min(\max(y, -20), 20))$, and l_0 the initial position of object I. P_F models the case where the robot starts in position l_0 and every move event changes the location of the robot on the field.

Robots R_1 and R_2 are two instances of the robot component, where all occurrences of R have been renamed respectively to R_1 and R_2 (e.g., $(read(loc, R), l)$ becomes $(read(loc, R_1), l)$ for the robot instance R_1, etc.). Similarly, we consider B_1 and B_2 to be two instances of the battery component B, and $F_1((0; 0))$ and $F_2((5; 0))$ to be two instances of the field component F parametrized by the initial location for the object I, where the objects in fields F_1 and F_2 are renamed to I_1 and I_2, and respectively initialized at position $(0; 0)$ and $(5; 0)$.

4.2. Interaction

We detail three points of interactions on observables among a robot and its battery, a robot and a field on which it moves, and two instances of a field component. The composability relations that relate the events of a robot, a battery, and a field impose some necessary constraints for the physical consistency of the cyber-physical system. For instance, that the power requested by the robot must match the characteristic of the battery.

Robot-battery. Interactions between a robot component and its battery are such that, for instance, every occurrence of a move event at the robot component must be simultaneous with a discharge event of the battery, with the discharge factor proportional to the demand of energy from the robot. Given a robot component R and a battery component B, we define the symmetric relation $\sqcap_{RB}$ on the set $\mathcal{P}(E_R \cup E_B)$ to be the smallest relation such that:

$$\begin{array}{rcll} \{(read(bat, R); b)\} & \sqcap_{RB} & \{(read(B); b)\} & \text{for all } 0 \leq b \leq C \\ \{(move(R); (d, \alpha))\} & \sqcap_{RB} & \{(discharge(B); \eta_d)\} & \text{for all } d \in \{\mathsf{N}, \mathsf{S}, \mathsf{W}, \mathsf{E}\} \\ \{(charge(R); \mathsf{ON})\} & \sqcap_{RB} & \{(charge(B); \eta_c)\} & \end{array}$$

with $\eta_d(t) > \alpha$ for all $t \in \mathbb{R}_+$, i.e., the power delivered by the battery during a discharge is greater than the power required by the move; and with C the capacity of the battery.

Robot-field. Interactions between a robot component and a field component are such that, for instance, every move event of the robot component must be simultaneous with a move event of the object I on the field, with a variable friction coefficient. Given a robot component R and a field component F, we define the symmetric relation $\sqcap_{RF}$ on the set $\mathcal{P}(E_R \cup E_F)$ to be the smallest relation such that:

$$\begin{array}{rcll} \{(read(loc, R); l)\} & \sqcap_{FR} & \{(loc(I); l)\} & \text{for all } l \in [0, 20]^2 \\ \{(move(R); (d, \alpha))\} & \sqcap_{FR} & \{(move(I); (d, F_t))\} & \text{for all } d \in \{\mathsf{N}, \mathsf{W}, \mathsf{E}, \mathsf{S}\}, v \in \mathbb{R}_+ \\ \{(charge(R); \mathsf{ON})\} & \sqcap_{FR} & \{(loc(I); (5, 5))\} & \end{array}$$

with $F_t = \dfrac{\alpha}{R\omega}$ with R the radius of the wheels of the robot and ω the speed of rotation of the wheels (assumed to be constant during the move). Observe that a robot can charge only if it is located at the charging station.

Field-field. We add also interaction constraints between two fields, such that no observation can gather two read events containing the same position value. Thus, given two fields F_1 and F_2, let $\sqcap_{F_{12}}$ be the smallest symmetric mutual exclusion relation on the set $\mathcal{P}(E_{F_1} \cup E_{F_2})$ such that:

$$\{(loc(I_1); l)\} \sqcap_{F_{12}} \{(loc(I_2); l)\} \text{ for all } l \in [0, 20]^2.$$

Observe that we interpret $\sqcap_{F_{12}}$ as a mutual exclusion relation. At first sight, the field does not prevent the two robots to share the same location. It only removes the possibility to observe the two robots at the same position. If, however, the field's behavior is closed under insertion of simultaneous read observables from both robots, then the two propositions stated above coincide (see Example 22).

Product. We use set union as a composition function on observables: given two observables O_1 and O_2, we define $O_1 \oplus O_2$ to be the observable $O_1 \cup O_2$. We use the synchronous and mutual exclusion composability relations on TESs introduced in Definition 9 and Definition 10. We represent the cyber-physical system consisting of two robots R_1 and R_2 with two private batteries B_1 and B_2, and individual fields F_1 and F_2, as the expression:

$$System = (F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2) \times_{[\kappa^{sync}_{(\sqcap F_1R_1 \cup \sqcap F_2R_2)}]} ((R_1 \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1) \times_{[\kappa_\top]} (R_2 \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2)) \tag{2}$$

Note that the previous expression describes the same component as:

$$System = ((F_1 \times_{[\kappa^{sync}_{\sqcap F_1R_1}]} R_1) \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1) \times_{[\kappa^{excl}_{\sqcap F_{12}}]} ((F_2 \times_{[\kappa^{sync}_{\sqcap F_2R_2}]} R_2) \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2) \tag{3}$$

Proof.

$$(F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2) \times_{[\kappa^{sync}_{(\sqcap F_1R_1 \cup \sqcap F_2R_2)}]} ((R_1 \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1) \times_{[\kappa_\top]} (R_2 \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2)) \tag{1}$$

$$= (F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2) \times_{[\kappa^{sync}_{(\sqcap F_1R_1 \cup \sqcap F_2R_2)}]} ((R_1 \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1) \times_{[\kappa^{sync}_{\sqcap F_{11} \cup \sqcap F_{22}}]} (R_2 \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2)) \tag{2}$$

$$= ((F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2) \times_{[\kappa^{sync}_{(\sqcap F_1R_1 \cup \sqcap F_2R_2)}]} (R_1 \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1)) \times_{[\kappa^{sync}_{\sqcap F_{11} \cup \sqcap F_{22}}]} ((F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2) \times_{[\kappa^{sync}_{(\sqcap F_1R_1 \cup \sqcap F_2R_2)}]} (R_2 \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2)) \tag{3}$$

$$= ((F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2) \times_{[\kappa^{sync}_{\sqcap F_1R_1}]} (R_1 \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1)) \times_{[\kappa^{sync}_{\sqcap F_{11} \cup \sqcap F_{22}}]} ((F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2) \times_{[\kappa^{sync}_{\sqcap F_2R_2}]} (R_2 \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2)) \tag{4}$$

$$= ((R_1 \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1) \times_{[\kappa^{sync}_{\sqcap F_1R_1}]} (F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2)) \times_{[\kappa^{sync}_{\sqcap F_{11} \cup \sqcap F_{22}}]} ((F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2) \times_{[\kappa^{sync}_{\sqcap F_2R_2}]} (R_2 \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2)) \tag{5}$$

$$= (R_1 \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1) \times_{[\kappa^{sync}_{\sqcap F_1R_1}]} (F_1 \times_{[\kappa^{excl}_{\sqcap F_{12}}]} F_2) \times_{[\kappa^{sync}_{\sqcap F_2R_2}]} (R_2 \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2) \tag{6}$$

$$= ((F_1 \times_{[\kappa^{sync}_{\sqcap F_1R_1}]} R_1) \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1) \times_{[\kappa^{excl}_{\sqcap F_{12}}]} ((F_2 \times_{[\kappa^{sync}_{\sqcap F_2R_2}]} R_2) \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2) \tag{7}$$

where:

- (1) to (2) is given by the fact that $((R_1 \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1) \times_{[\kappa_\top]} (R_2 \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2)) = ((R_1 \times_{[\kappa^{sync}_{\sqcap R_1B_1}]} B_1) \times_{[\kappa^{sync}_{\sqcap F_{11} \cup \sqcap F_{22}}]} (R_2 \times_{[\kappa^{sync}_{\sqcap R_2B_2}]} B_2))$, i.e., synchronization on events that are not in the interface is the same as the relation $\kappa_\top$ (Lemma 10);
- (2) to (3) is given by Lemma 9;
- (3) to (4) simplifies in both side the synchronous composability relation to range over the interface of its operand components;
- (4) to (5) commutativity of $\times_{[\kappa^{sync}_{\sqcap F_1R_1}]}$;
- (5) to (6) first rewrite $\kappa^{sync}_{\sqcap F_1R_1}$, $\kappa^{sync}_{\sqcap F_2R_2}$, and $\kappa^{sync}_{\sqcap F_{11} \cup \sqcap F_{22}}$ to all the same $\kappa^{sync}_{\sqcap}$ where $\sqcap = \sqcap_{F_{11}} \cup \sqcap_{F_{22}} \cup \sqcap_{F_1R_1} \cup \sqcap_{F_2R_2}$ which does not change the synchronization product (Lemma 10); and then uses associativity and idempotency of the product $\times_{[\kappa^{sync}_{\sqcap}]}$;
- (6) to (7) the synchronous products are distributed to the component on which they have effect only, i.e., F_1 and F_2 for R_1 and R_2 respectively. □

4.3. Behavioral properties of components

Let $E = E_{R_1} \cup E_{R_2} \cup E_{B_1} \cup E_{B_2} \cup E_{F_1} \cup E_{F_2}$ be the set of events for the composite system in Equation (2). We formulate the scenarios described in Section 2 in terms of a satisfaction problem involving a safety property on TESs and a behavior property on the composite system. We first consider two safety properties:

$$P_{energy} = \{\sigma \in TES(E) \mid \forall i \in \mathbb{N}.\{(read(B_1), 0), (read(B_2), 0)\} \cap \mathrm{pr}_1(\sigma)(i) = \emptyset\}$$

$$P_{no-overlap} = \{\sigma \in TES(E) \mid \forall i \in \mathbb{N}.\forall l \in [0, 20]^2, \{(loc(I_1), l), (loc(I_2), l)\} \nsubseteq \mathrm{pr}_1(\sigma)(i)\}$$

The property P_{energy} collects all behaviors that never observe a battery value of 0 Wh. The property $P_{no-overlap}$ describes all behaviors where the two robots are never observed together at the same location. Observe that, while both P_{energy} and $P_{no-overlap}$ specify some safety properties, they are not sufficient to ensure the safety of the system. We illustrate some scenarios with the property P_{energy}. If a component never reads its battery level, then the property P_{energy} is trivially satisfied, although effectively the battery may run out of energy. Also, if a component reads its battery level periodically, each of its readings may return an observation agreeing with the property. However, in between two read events, the battery may run out of energy (and somehow recharge). To circumvent those unsafe scenarios, we add an additional behavioral property.

Let $X_{read} = \{(read(B_1); l_1), (read(B_2); l_2) \mid 0 \le l_1 \le C_1,\ 0 \le l_2 \le C_2\}$ be the set of reading events for battery components B_1 and B_2, with maximal charge C_1 and C_2 respectively. The property $\phi_{insert}(X_{read}, E)$, as detailed in Example 23, defines a class of component behaviors that are closed under insertion of *read* events for the battery component. Therefore, the system denoted as C, defined in Equation (2) is energy safe if $C \models P_{energy}$ and its behavior is closed under insertion of battery read events, i.e., $C \| \models \phi_{insert}(X_{read}, E)$. In that case, every TES of the component's behavior is part of a set that is

closed under insertion, which means all read events that the robot may do in between two events observe a battery level greater than 0 Wh. The behavior property enforces the following safety principle: had there been a violating behavior (i.e., a run where the battery has no energy), then an underlying TES would have observed it, and hence the behavioral property would have been violated.

Another scenario for the two robots is to consider their coordination in order to have them swap their positions. Let F_1 be initialized to have object I_1 at position $(0,0)$ and F_2 have I_2 at position $(5,0)$. The property of position swapping is a liveness property defined as:

$$\mathrm{P}_{\mathrm{swap}} = \{\sigma \in \mathit{TES}(E \cup \{\diamond\}) \mid \{(\mathit{loc}(I_1),(0,0)),(\mathit{loc}(I_2),(5,0))\} \subseteq \mathrm{pr}_1(\sigma)(0) \text{ and } \exists i \in \mathbb{N}.\{(\mathit{loc}(I_1),(5,0)),(\mathit{loc}(I_2),(0,0)),\diamond\} \subseteq \mathrm{pr}_1(\sigma)(i)\}$$

where $\diamond$ is used as an external symbol not in E. It is sufficient for a liveness property to be satisfied for the system to be live, i.e., in the case of $\mathrm{P}_{\mathrm{swap}}$, eventually reach a swapped position. However, it may be that the two robots swap their positions before the actual observation happens. In that case, using a similar behavioral property as for safety property will make sure that if there exists a behavior where robots swap their positions, then such behavior is observed as soon as it happens.

Given a set of events E, let $\sqcap_E = \{(O,O) \mid O \subseteq E\}$ be a relation on sets of events. Let $\mathrm{P}_{\mathrm{swap}} \downarrow R_i \subseteq \mathit{TES}(E_{R_i} \cup \{\diamond\})$ be the projection of property $\mathrm{P}_{\mathrm{swap}}$ on the set of events E_{R_i} such that:

$$\tau \in \mathrm{P}_{\mathrm{swap}} \downarrow R_i \iff \exists \sigma \in \mathrm{P}_{\mathrm{swap}}.\forall n \in \mathbb{N}.(\sigma(n) = (O,t) \implies \tau(n) = (O \cap (E_{R_i} \cup \{\diamond\}),t))$$

Let $C^{R_i}_{\mathrm{swap}} = (E_{R_i} \cup \{\diamond\}, \mathrm{P}_{\mathrm{swap}} \downarrow R_i)$ and $C_{\mathrm{swap}} = (E_{R_1} \cup E_{R_2} \cup \{\diamond\}, \mathrm{P}_{\mathrm{swap}})$, then $C_{\mathrm{swap}} = (C^{R_i}_{\mathrm{swap}} \times_{\kappa^{sync}_{\sqcap_{\{\diamond\}}}} C^{R_i}_{\mathrm{swap}})$. We show an equivalence between coordination of the two robots by a centralized coordinator (e.g., C_{swap}), and coordination of the two robots by a decentralized coordination (e.g., $C^{R_1}_{\mathrm{swap}}$ and $C^{R_2}_{\mathrm{swap}}$).

Indeed, due to Lemma 9:

$$C_{\mathrm{swap}} \times_{[\kappa^{sync}_{\sqcap_{E_{R_1} \cup \sqcap E_{R_2}}}]} (R_1 \times_{[\kappa^{sync}_{\top}]} R_2) = (C^{R_1}_{\mathrm{swap}} \times_{[\kappa^{sync}_{\sqcap_{\{\diamond\}}}]} C^{R_2}_{\mathrm{swap}}) \times_{[\kappa^{sync}_{\sqcap_{E_{R_1} \cup \sqcap E_{R_2}}}]} (R_1 \times_{[\kappa^{sync}_{\top}]} R_2) \tag{8}$$

$$= (C^{R_1}_{\mathrm{swap}} \times_{[\kappa^{sync}_{\sqcap_{E_{R_1}}}]} R_1) \times_{[\kappa^{sync}_{\sqcap_{\{\diamond\}}}]} (C^{R_2}_{\mathrm{swap}} \times_{[\kappa^{sync}_{\sqcap_{E_{R_2}}}]} R_2) \tag{9}$$

The above example shows how a property can be decomposed into sub-properties that interact via some shared events. Such decomposition makes, a structural distinction between a *local* form of coordination imposed by the products of the sub-property and its interacting component, and a *global* form of coordination that coordinate the two properties using an additional $\diamond$ signal. Moreover, this scheme demonstrates the ability to localize a coordinator next to the components that it coordinates, which makes our design modular.

5. Related and future work

Our work offers a component-based semantics for cyber-physical systems [12,13]. In [14], a similar aim is pursued by defining an algebra of components using interface theory. Our component-based approach is inspired by [3,15], where a component exhibits its behavior as a set of infinite timed-data streams. More details about co-algebraic techniques to prove component equivalences can be found in [16].

Our model of components assumes some underlying physical models. We do not give a precise account on how to model physics of a component, in contrast to work using hybrid models [17]. TESs contrast to discrete event modes, such as the work in [18,5,19], by being based on arbitrary time sampling strategies rather than event driven observations. We abstract and generalize such work by supporting time sensitive system evolution and more generally observation of physical properties.

In [8], the authors describe an algebra of timed machines and of networks of timed machines. A timed machine is a state based description of a set of timed traces, such that every observation has a time stamp that is a multiple of a time step δ. This work differs from our current development in several respects. We focus in this paper on different algebraic operations on sets of timed-traces (TESs), and abstract away any underlying operational model (e.g., timed-automata). In [8], the authors explain how algebraic operations on timed machines *approximate* the intersection of sets of timed-traces. In our case, interaction is not restricted to input/output composition, but depends on the choice of a composability constraint on TESs and a composition function on observables. The work in [8] defines an interesting class of components (closed under insertion of silent observation - r-closed) that deserves investigation.

Cyber-physical systems have also been studied from an actor-model perspective, where actors interact through events [20]. Methods for achieving synchronous behavior using asynchronous means of interaction are presented in [21].

In [22] a multiset rewriting model of time sensitive distributed systems such as cyber-physical agent, is introduced. Two verification problems are defined relative to a given property P: realizability (is there a trace that satisfies P), and

survivability (do all traces satisfy P) and their complexity is analyzed. In [23] the theory is extended with two further properties that concern the ability to avoid reaching a bad state.

Hybrid Communicating Sequential Processes (HCSP) [24] is an extension of CSP [25] in order to model communicating hybrid systems. In [26], the authors give a semantics using Higher-order Unifying Theories of Programming (HUTP). The communicating behavior of hybrid processes is captured by a timed trace model, where elements of a timed trace are either communication blocks, wait blocks, or internal blocks. This model differs from our semantic model as we consider only observable behavior and no internal or waiting behavior.

Recent work has shown plenty of interest in studying the satisfaction problem of hyper-properties and the synthesis of reactive systems [27]. Some works focus more particularly on using hyper-properties for cyber-physical design [28].

The extension of hybrid automata [29] to quantized hybrid automata is presented in [30], where the authors apply their model to give a formal semantics for data flow models of cyber-physical systems such as Simulink [31].

Compared to formalisms that model cyber-physical systems as more concrete operational or state-based mechanisms, such as automata or abstract machines, our more general abstract formalism is based only on the observable behavior of cyber-physical components and their composition into systems, regardless of what more concrete models or mechanisms may produce such behavior.

For future work, we want to provide a finite description for components, and use our current formalism as its formal semantics. In fact, we first started to model interactive cyber-physical systems as a set of finite state automata in composition, but the underlying complexity of automata interaction led us to introduce a more abstract component model to clarify the semantics of those interactions. Moreover, we want to investigate several proof techniques to show equivalences of components. We expect to be able to reason about local and global coordination, by studying how coordinators distribute over our different composition operators. Finally, our current work serves as a basis for defining a compositional semantics for a state-based component framework [32] written in Maude [33], a specification and programming language based on rewriting logic. We plan to focus on evaluating the robustness of a set of components with respect to system requirements expressed as trace- or hyper-properties. The complexity of the satisfaction problem requires some run-time techniques to detect deviations and produce meaningful diagnoses [11], a topic that we are currently exploring.

6. Conclusion

This paper contains three main contributions. First, we introduce a component model for cyber-physical systems where cyber and physical processes are uniformly described in terms of sequences of observations. Second, we provide ways to express interaction among components using algebraic operations, such as a parametric product and division, and give conditions under which product is associative, commutative, or idempotent. Third, we provide a formal basis to study trace- and hyper-properties of components, and demonstrate the application of our work in an example describing several coordination problems.

Our semantic model provides a formal basis for designing interacting cyber-physical systems, where interaction is defined explicitly and exogenously as an algebraic operation acting on components. As a future step, we plan to use the semantic model introduced in this work to give a compositional semantics for interacting (state-based) specification for cyber-physical components. We aim to use our modular design in order to study problems of diagnosis in systems of interacting cyber-physical components.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Benjamin Lion reports financial support was provided by National Research Institute for Mathematics and Computer Science (CWI). Carolyn Talcott reports a relationship with SRI International that includes. Farhad Arbab reports a relationship with National Research Institute for Mathematics and Computer Science that includes (CWI).

Acknowledgement

We thank the anonymous reviewers for their careful reading and their insightful comments. Talcott was partially supported by the U.S. Office of Naval Research under award numbers N00014-15-1-2202 and N00014-20-1-2644, and NRL grant N0017317-1-G002. Arbab was partially supported by the U.S. Office of Naval Research under award number N00014-20-1-2644.

Appendix A. Proof

Proof of Lemma 7. Commutativity. From Lemma 3, if κ is symmetric, then its lifting $[\kappa]$ is also symmetric. Therefore, it is sufficient for κ to be symmetric and for $+$ to be commutative in order for $[\kappa]$ to be symmetric and $[+]$ to be commutative, and therefore $\times_{([\kappa],[+])}$ to be commutative.

Associativity. A sufficient condition for the product $\times_{([\kappa],[+])}$ to be associative is that $+$ is associative and for every $\sigma_i \in TES(E_i)$ for $i \in \{1, 2, 3\}$:

$$P_1 := \begin{array}{l}(\sigma_1,\sigma_2) \in [\kappa](E_1,E_2) \wedge (\sigma_1[+]\sigma_2,\sigma_3) \in [\kappa](E_1 \cup E_2, E_3) \iff \\ (\sigma_2,\sigma_3) \in [\kappa](E_2,E_3) \wedge (\sigma_1,\sigma_2[+]\sigma_3) \in [\kappa](E_1, E_2 \cup E_3)\end{array}$$

We introduce the function

$$\Psi^1_\kappa(E_1,E_2,E_3)(\mathcal{R}) = \{(\sigma_1,\sigma_2,\sigma_3) \mid \begin{array}{l}(\sigma_1(0),\sigma_2(0)) \in \kappa(E_1,E_2) \wedge ((\sigma_1[+]\sigma_2)(0),\sigma_3(0)) \in \kappa(E_1 \cup E_2, E_3) \wedge \\ (\sigma_1,\sigma_2,\sigma_3)' \in \mathcal{R})\}\end{array}$$

and

$$\Psi^2_\kappa(E_1,E_2,E_3)(\mathcal{R}) = \{(\sigma_1,\sigma_2,\sigma_3) \mid \begin{array}{l}(\sigma_2(0),\sigma_3(0)) \in \kappa(E_2,E_3) \wedge (\sigma_1(0),(\sigma_2[+]\sigma_3)(0)) \in \kappa(E_1, E_2 \cup E_3) \wedge \\ (\sigma_1,\sigma_2,\sigma_3)' \in \mathcal{R})\}\end{array}$$

where $(\sigma_1,\sigma_2,\sigma_3)' = (\sigma_1^{(x)},\sigma_2^{(y)},\sigma_3^{(z)})$ with $(x,y,z) = \tau(1)$ where τ is the enumeration of $(\sigma_1,\sigma_2,\sigma_3)$ as introduced in Definition 12. Showing that P_1 holds is equivalent to showing that $\mathcal{R} \subseteq \Psi^1_\kappa(E_1,E_2,E_3)(\mathcal{R})$ if and only if $\mathcal{R} \subseteq \Psi^2_\kappa(E_1,E_2,E_3)(\mathcal{R})$ for any $\mathcal{R} \subseteq TES(E_1) \times TES(E_2) \times TES(E_3)$.

First, we observe that, given $\mathcal{R} \subseteq \Psi^1_\kappa(E_1,E_2,E_3)(\mathcal{R})$, for any $(\sigma_1,\sigma_2,\sigma_3) \in \mathcal{R}$ and its enumeration τ, for all $n \in \mathbb{N}$ letting $\tau(n) = (i,j,k)$, we have:

$$(\sigma_1(i),\sigma_2(j)) \in \kappa(E_1,E_2) \wedge ((\sigma_1^{(i)}[+]\sigma_2^{(j)})(0),\sigma_3(k)) \in \kappa(E_1 \cup E_2, E_3)$$

Similarly, if the same $\mathcal{R}$ is such that $\mathcal{R} \subseteq \Psi^2_\kappa(E_1,E_2,E_3)(\mathcal{R})$, then, for all n there exist i,j,k such that $\tau(n) = (i,j,k)$ and

$$(\sigma_2(j),\sigma_3(k)) \in \kappa(E_2,E_3) \wedge (\sigma_1(i),(\sigma_2^{(j)}[+]\sigma_3^{(k)})(0)) \in \kappa(E_1, E_2 \cup E_3)$$

Thus, to show that P_1 holds, it is sufficient to prove that, for all $n \in \mathbb{N}$ there exists (i,j,k) with $\tau(n) = (i,j,k)$ and

$$(\sigma_1(i),\sigma_2(j)) \in \kappa(E_1,E_2) \wedge ((\sigma_1^{(i)}[+]\sigma_2^{(j)})(0),\sigma_3(k)) \in \kappa(E_1 \cup E_2, E_3)$$

if and only if, for all $n \in \mathbb{N}$ there exists (i,j,k) with $\tau(n) = (i,j,k)$ and

$$(\sigma_2(j),\sigma_3(k)) \in \kappa(E_2,E_3) \wedge (\sigma_1(i),(\sigma_2^{(j)}[+]\sigma_3^{(k)})(0)) \in \kappa(E_1, E_2 \cup E_3)$$

The above equivalence holds for any $(\sigma_1,\sigma_2,\sigma_3) \in TES(E_1) \times TES(E_2) \times TES(E_3)$ with τ the enumeration of $(\sigma_1,\sigma_2,\sigma_3)$.

Finally, we prove that if $+$ is associative, then $[+]$ is associative. Let $\sigma_i \in L_i$ and we write $\sigma_i(0) = (O_i,t_i)$ for $i \in \{1,2,3\}$, then:

$$\sigma_1[+](\sigma_2[+]\sigma_3) = \begin{cases}\langle(O_1,t_1)\rangle \cdot (\sigma_1'[+](\sigma_2[+]\sigma_3) & \text{if } t_1 < t_2 \wedge t_1 < t_3 \\ \langle(O_2,t_2)\rangle \cdot (\sigma_1[+](\sigma_2'[+]\sigma_3) & \text{if } t_2 < t_1 \wedge t_2 < t_3 \\ \langle(O_3,t_3)\rangle \cdot (\sigma_1[+](\sigma_2[+]\sigma_3') & \text{if } t_3 < t_2 \wedge t_3 < t_1 \\ \langle(O_1+O_2,t_1)\rangle \cdot (\sigma_1'[+](\sigma_2'[+]\sigma_3) & \text{if } t_1 = t_2 \wedge t_1 < t_3 \\ \langle(O_2+O_3,t_2)\rangle \cdot (\sigma_1[+](\sigma_2'[+]\sigma_3') & \text{if } t_2 = t_3 \wedge t_2 < t_1 \\ \langle(O_1+O_3,t_1)\rangle \cdot (\sigma_1'[+](\sigma_2[+]\sigma_3') & \text{if } t_1 = t_3 \wedge t_1 < t_2 \\ \langle(O_1+(O_2+O_3),t_1)\rangle \cdot (\sigma_1'[+](\sigma_2'[+]\sigma_3') & \text{if } t_1 = t_3 \wedge t_1 = t_2\end{cases}$$

The only case that differs from $(\sigma_1[+]\sigma_2)[+]\sigma_3$ is when $t_1 = t_3 = t_2$, which gives $((O_1+O_2)+O_3,t_1)$. Thus, if $((O_1+O_2)+O_3,t_1) = (O_1+(O_2+O_3),t_1)$ for every $O_i \in \mathcal{P}(E_i)$ with $i \in \{1,2,3\}$, then $\sigma_1[+](\sigma_2[+]\sigma_3) = \sigma_1[+](\sigma_2[+]\sigma_3)$ for every $\sigma_i \in L_i$ with $i \in \{1,2,3\}$.

Idempotency. If $+$ is idempotent, then the lifting $[+]$ is also idempotent. We consider $+$ to be idempotent. We show that, for all $E \subseteq \mathbb{E}$ and $o_1,o_2 \in \mathcal{P}(E) \times \mathbb{R}_+$ we have $(o_1,o_2) \in \kappa(E,E) \implies o_1 = o_2$, then for all $\sigma,\tau \in TES(E)$, $(\sigma,\tau) \in [\kappa](E,E) \implies \sigma = \tau$, which is a sufficient condition for $\times_{([\kappa],[+])}$ to be idempotent.

By definition $[\kappa](E,E)$ is the greatest fixed point of the function:

$$\begin{array}{rl}\Phi_\kappa(E,E)(\mathcal{R}) = \{(\tau_1,\tau_2) \mid & (\tau_1(0),\tau_2(0)) \in \kappa(E,E) \wedge (\tau_1,\tau_2)' \in \mathcal{R})\} \\ \subseteq \{(\tau_1,\tau_2) \mid & \tau_1(0) = \tau_2(0) \wedge (\tau_1',\tau_2') \in \mathcal{R}\}\end{array}$$

Therefore, we conclude that $[\kappa](E,E) \subseteq \{(\sigma,\sigma) \mid \sigma \in TES(E)\}$. □

Proof of Lemma 8. Let $E_1,E_2,E_3 \subseteq \mathbb{E}$, and $\sqcap \subseteq \mathcal{P}(E_1 \cup E_2 \cup E_3) \times \mathcal{P}(E_1 \cup E_2 \cup E_3)$. We use $\kappa^{sync}_\sqcap(E_1,E_2)$, the composability relation in Definition 9:

- for all $(O_1,O_2) \in \mathcal{P}(E_1) \times \mathcal{P}(E_2)$ such that $(O_1,O_2) \in \sqcap$ and for all $(O_1',O_2') \in \mathcal{P}(E_1) \times \mathcal{P}(E_2)$ such that $ind_\sqcap(O_1',E_2)$ and $ind_\sqcap(E_1,O_2')$ then, for all time stamps t, $((O_1 \cup O_1',t),(O_2 \cup O_2',t)) \in \kappa^{sync}_\sqcap(E_1,E_2)$.
- if $ind_\sqcap(O_1,E_2)$ then for all $O_2 \subseteq E_2$ and $t_1 \preceq t_2$, $((O_1,t_1),(O_2,t_2)) \in \kappa^{sync}_\sqcap(E_1,E_2)$. Reciprocally, if $ind_\sqcap(E_1,O_2)$ then for all $O_1 \subseteq E_1$ and $t_2 \preceq t_1$, $((O_1,t_1),(O_2,t_2)) \in \kappa^{sync}_\sqcap(E_1,E_2)$;

where $ind_\sqcap(X, Y) = \forall x \subseteq X.\ \forall y \subseteq Y.\ (x, y) \notin \sqcap$.

Let $(\sigma_1, \sigma_2, \sigma_3) \in TES(E_1) \times TES(E_2) \times TES(E_3)$ and let τ be an enumeration of $(\sigma_1, \sigma_2, \sigma_3)$. We show that, if, for all $n \in \mathbb{N}$, letting $\tau(n) = (i, j, k)$, the following holds

$$(\sigma_1(i), \sigma_2(j)) \in \kappa_\sqcap^{sync}(E_1, E_2) \wedge ((\sigma_1^{(i)}[\cup]\sigma_2^{(j)})(0), \sigma_3(k)) \in \kappa_\sqcap^{sync}(E_1 \cup E_2, E_3)$$

then, for all $n \in \mathbb{N}$, letting $\tau(n) = (i, j, k)$, the following holds

$$(\sigma_2(j), \sigma_3(k)) \in \kappa_\sqcap^{sync}(E_2, E_3) \wedge (\sigma_1(i), (\sigma_2^{(j)}[\cup]\sigma_3^{(k)})(0)) \in \kappa_\sqcap^{sync}(E_1, E_2 \cup E_3)$$

(the implication in the other direction is similar). We proceed by induction on n (i.e., the elements of the enumeration) in the right hand side of the implication.

Base case. Let $n = 0$. Then, $\tau(n) = \tau(0) = (0, 0, 0)$. We show that

$$(\sigma_2(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_2, E_3) \wedge (\sigma_1(0), (\sigma_2[\cup]\sigma_3)(0)) \in \kappa_\sqcap^{sync}(E_1, E_2 \cup E_3)$$

We know that, for all n, letting $\tau(n) = (i, j, k)$, we have

$$(\sigma_1(i), \sigma_2(j)) \in \kappa_\sqcap^{sync}(E_1, E_2) \wedge ((\sigma_1^{(i)}[\cup]\sigma_2^{(j)})(0), \sigma_3(k)) \in \kappa_\sqcap^{sync}(E_1 \cup E_2, E_3)$$

Thus, for $n = 0$, we know that

$$(\sigma_1(0), \sigma_2(0)) \in \kappa_\sqcap^{sync}(E_1, E_2) \wedge ((\sigma_1[\cup]\sigma_2)(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_1 \cup E_2, E_3)$$

Let $\sigma_1(0) = (O_1, t_1)$, $\sigma_2(0) = (O_2, t_2)$, and $\sigma_3(0) = (O_3, t_3)$, we split cases on time stamp values and show the implication for each of those cases.

- Let $t_1 < t_2 \wedge t_1 < t_3$. Then

  $$(\sigma_1(0), \sigma_2(0)) \in \kappa_\sqcap^{sync}(E_1, E_2) \wedge (\sigma_1(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_1 \cup E_2, E_3)$$

  which implies that $(\sigma_1(0), (\sigma_2[\cup]\sigma_3)(0)) \in \kappa_\sqcap^{sync}(E_1, E_2 \cup E_3)$ by the fact that $ind_\sqcap(O_1, E_2)$ and $ind_\sqcap(O_1, E_3)$ together imply that $ind_\sqcap(O_1, E_2 \cup E_3)$. Then, one can show that, given the property of the enumeration, there exists an i' such that $\tau(i') = (i', 0, 0)$ with $t_2 \leq \mathrm{pr}_2(\sigma_1)(i')$ or $t_3 \leq \mathrm{pr}_2(\sigma_1)(i')$. We discuss those cases below to prove that $(\sigma_2(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_2, E_3)$.
- Let $t_2 < t_1 \wedge t_2 < t_3$. Then,

  $$(\sigma_1(0), \sigma_2(0)) \in \kappa_\sqcap^{sync}(E_1, E_2) \wedge (\sigma_2(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_1 \cup E_2, E_3)$$

  The definition of $\kappa_\sqcap^{sync}$ implies that, if $((O, t), (O', t')) \in \kappa^{sync}(E, E')$ and $t < t'$ then $((O, t), (O', t')) \in \kappa_\sqcap^{sync}(E'', E')$ for arbitrary E'' with $O \subseteq E''$. Symmetrically, if $t' < t$, then $((O, t), (O', t')) \in \kappa_\sqcap^{sync}(E, E'')$ for arbitrary E'' with $O' \subseteq E''$. Thus,

  $$(\sigma_2(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_2, E_3) \wedge (\sigma_1(0), \sigma_2(0)) \in \kappa_\sqcap^{sync}(E_1, E_2 \cup E_3)$$
- Let $t_3 < t_2 \wedge t_3 < t_1$. Then,

  $$(\sigma_1(0), \sigma_2(0)) \in \kappa_\sqcap^{sync}(E_1, E_2) \wedge ((\sigma_1[\cup]\sigma_2)(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_1 \cup E_2, E_3)$$

  By definition of $ind_\sqcap$, if $ind_\sqcap(E_1 \cup E_2, O_3)$, then $ind_\sqcap(E_2, O_3)$ and $ind_\sqcap(E_1, O_3)$. Thus,

  $$(\sigma_2(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_2, E_3) \wedge (\sigma_1(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_1, E_2 \cup E_3)$$
- Let $t_1 = t_2 \wedge t_1 < t_3$. Then,

  $$(\sigma_1(0), \sigma_2(0)) \in \kappa_\sqcap^{sync}(E_1, E_2) \wedge ((\sigma_1[\cup]\sigma_2)(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_1 \cup E_2, E_3)$$

  By definition of $\kappa_\sqcap^{sync}$, there exists $O_1' \subseteq O_1$ and $O_2' \subseteq O_2$ such that $(O_1', O_2') \in \sqcap$ with $ind_\sqcap(O_1 \setminus O_1', E_2)$ and $ind_\sqcap(E_1, O_2 \setminus O_2')$. Moreover, $ind_\sqcap(O_1 \cup O_2, E_3)$ implies that $ind_\sqcap(O_2, E_3)$ and $ind_\sqcap(O_1, E_3)$ using the definition of $ind_\sqcap$. Thus,

  $$(\sigma_2(0), \sigma_3(0)) \in \kappa_\sqcap^{sync}(E_2, E_3) \wedge (\sigma_1(0), \sigma_2(0)) \in \kappa_\sqcap^{sync}(E_1, E_2 \cup E_3)$$

- Let $t_2 = t_3 \wedge t_2 < t_1$. Then,

$$(\sigma_1(0), \sigma_2(0)) \in \kappa_{\sqcap}^{sync}(E_1, E_2) \wedge (\sigma_2(0), \sigma_3(0)) \in \kappa_{\sqcap}^{sync}(E_1 \cup E_2, E_3)$$

  Similar arguments as for the case where $t_1 = t_2 \wedge t_1 < t_3$ help to conclude that

$$(\sigma_2(0), \sigma_3(0)) \in \kappa_{\sqcap}^{sync}(E_2, E_3) \wedge (\sigma_1(0), (\sigma_2[\cup]\sigma_3)(0)) \in \kappa_{\sqcap}^{sync}(E_1, E_2 \cup E_3)$$

- Let $t_1 = t_3 \wedge t_1 < t_2$. Then,

$$(\sigma_1(0), \sigma_2(0)) \in \kappa_{\sqcap}^{sync}(E_1, E_2) \wedge (\sigma_1(0), \sigma_3(0)) \in \kappa_{\sqcap}^{sync}(E_1 \cup E_2, E_3)$$

  By definition of $\kappa_{\sqcap}^{sync}$, there exist $O_1' \subseteq O_1$ and $O_3' \subseteq O_3$ such that $(O_1', O_3') \in \sqcap$ with $ind_{\sqcap}(O_1 \setminus O_1', E_3)$ and $ind_{\sqcap}(E_1 \cup E_2, O_3 \setminus O_3')$ and $ind_{\sqcap}(O_1, E_2)$. Moreover, Property 1 item 1 together with $ind_{\sqcap}(O_1, E_2)$ and $ind_{\sqcap}(O_1 \setminus O_1', E_3)$, we can conclude that $ind_{\sqcap}(O_1 \setminus O_1', E_2 \cup E_3)$. Also, since $ind_{\sqcap}(O_1, E_2)$ and $(O_1', O_3') \in \sqcap$, it must be that $ind_{\sqcap}(E_2, O_3')$ otherwise, there would be an $O_2' \subseteq E_2$ with $(O_2', O_3') \in \sqcap$ and therefore $(O_1', O_2') \in \sqcap$ by Property 1 item 2. Thus, $ind_{\sqcap}(E_2, O_3)$, and

$$(\sigma_2(0), \sigma_3(0)) \in \kappa_{\sqcap}^{sync}(E_2, E_3) \wedge (\sigma_1(0), \sigma_3(0)) \in \kappa_{\sqcap}^{sync}(E_1, E_2 \cup E_3)$$

- Let $t_1 = t_3 \wedge t_1 = t_2$. Then,

$$(\sigma_1(0), \sigma_2(0)) \in \kappa_{\sqcap}^{sync}(E_1, E_2) \wedge ((\sigma_1[\cup]\sigma_2)(0), \sigma_3(0)) \in \kappa_{\sqcap}^{sync}(E_1 \cup E_2, E_3)$$

  By definition of $\kappa_{\sqcap}^{sync}$, there exist $O_1' \subseteq O_1$, $O_2' \subseteq O_2$ such that $(O_1', O_2') \in \sqcap$, $ind_{\sqcap}(O_1 \setminus O_1', E_2)$, and $ind_{\sqcap}(E_1, O_2 \setminus O_2')$; and there exist $O_1'' \subseteq O_1$, $O_2'' \subseteq O_2$, $O_3' \subseteq O_3$ such that $(O_1'' \cup O_2'', O_3') \in \sqcap$ and $ind_{\sqcap}(E_1, O_3 \setminus O_3')$; and $ind_{\sqcap}((O_1 \cup O_2) \setminus (O_1'' \cup O_2''), E_3)$. If $O_2'' = \emptyset$, then $ind_{\sqcap}(O_2, E_3)$ and $(\sigma_2(0), \sigma_3(0)) \in \kappa_{\sqcap}^{sync}(E_2, E_3)$; and if $O_2'' \neq \emptyset$, then, without loss of generality, we can assume that $\neg ind_{\sqcap}(O_2'', E_3)$ and that there exists $O_2''' \subseteq O_2''$ and $O_3'' \subseteq O_3'$ such that $(O_2''', O_3'') \in \sqcap$ and $ind_{\sqcap}(O_2 \setminus O_2''', E_3)$ and $ind_{\sqcap}(E_2, O_3 \setminus O_3'')$. Thus, $(\sigma_2(0), \sigma_3(0)) \in \kappa_{\sqcap}^{sync}(E_2, E_3)$.
  Similarly, there exists $O_1''' \subseteq O_1''$ and $O_3''' \subseteq O_3'$ such that $ind_{\sqcap}(O_1 \setminus O_1''', E_3)$, $ind_{\sqcap}(E_1, O_3 \setminus O_3''')$, and $(O_1''', O_3''') \in \sqcap$, and therefore $(O_1''' \cup O_1', O_3''' \cup O_2') \in \sqcap$ by Property 1 item 5. Using Property 1 item 1 with $ind_{\sqcap}(E_1, (O_2 \setminus O_2') \cup (O_3 \setminus O_3''))$, we get that $ind_{\sqcap}(E_1, (O_2 \cup O_3) \setminus (O_2' \cup O_3'''))$ that and together with $ind_{\sqcap}(O_1 \setminus (O_1''' \cup O_1'), E_2 \cup E_3)$, leads to the conclusion that $(\sigma_1(0), (\sigma_2[\cup]\sigma_3)(0)) \in \kappa_{\sqcap}^{sync}(E_1, E_2 \cup E_3)$.

We showed the base case, for $n = 0$.

Inductive argument. Let n be arbitrary, and $\tau(n) = (i, j, k)$. We show that the result holds for $n+1$ and $\tau(n+1) = (i', j', k')$. The inductive argument relies on the fact that one may apply the initialization arguments for the triple $(\sigma_1^{(i)}, \sigma_2^{(j)}, \sigma_3^{(k)})$ with enumeration $\tau^{(n)}$. Based on the above arguments, one can therefore prove that the implication holds for (i', j', k'), and conclude that it holds for every n, which proves the statement.

Reflexivity. If $\sqcap$ is co-reflexive, then $(O_1, O_2) \in \sqcap$ implies $O_1 = O_2$. Suppose any two observations $(O_1, t_1), (O_2, t_2)$ with $O_1, O_2 \subseteq E$. Due to the reflexivity of $\sqcap$, $(O_1, O_1) \in \sqcap$ and $(O_2, O_2) \in \sqcap$. It must be that $t_1 = t_2$. Moreover, there is no $O_1', O_2' \subseteq E$ such that $(O_1, O_2') \in \sqcap$ or $(O_1', O_2) \in \sqcap$. Thus, we can conclude that $((O_1, t_1), (O_2, t_2)) \in \kappa_{\sqcap}^{sync}(E, E)$ implies $O_1 = O_2$ and $t_1 = t_2$. We therefore conclude that the product $\times_{([\kappa_{\sqcap}^{sync}], [\cup])}$ is idempotent. □

References

[1] X. Liu, E. Matsikoudis, E.A. Lee, Modeling timed concurrent systems, in: C. Baier, H. Hermanns (Eds.), CONCUR 2006 – Concurrency Theory, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 1–15.
[2] S. Tripakis, C. Stergiou, C. Shaver, E.A. Lee, A modular formal semantics for Ptolemy, Math. Struct. Comput. Sci. 23 (2013) 834–881.
[3] F. Arbab, J.J.M.M. Rutten, A coinductive calculus of component connectors, in: M. Wirsing, D. Pattinson, R. Hennicker (Eds.), Recent Trends in Algebraic Development Techniques, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 34–55.
[4] G. Lafferriere, G.J. Pappas, S. Yovine, Symbolic reachability computation for families of linear vector fields, J. Symb. Comput. 32 (3) (2001) 231–253, https://doi.org/10.1006/jsco.2001.0472.
[5] S. Lafortune, Discrete event systems: modeling, observation, and control, in: Annual Review of Control, Robotics, and Autonomous Systems, 2019.
[6] M.R. Clarkson, F.B. Schneider, Hyperproperties, J. Comput. Secur. 18 (6) (2010) 1157–1210.
[7] B. Lion, F. Arbab, C. Talcott, A semantic model for interacting cyber-physical systems, Electron. Proc. Theor. Comput. Sci. 347 (2021) 77–95, https://doi.org/10.4204/eptcs.347.5.
[8] J. Fiadeiro, A. Lopes, B. Delahaye, A. Legay, Dynamic networks of heterogeneous timed machines, Math. Struct. Comput. Sci. 28 (6) (2018) 800–855, https://doi.org/10.1017/S0960129517000135.
[9] B. Alpern, F.B. Schneider, Defining liveness, Inf. Process. Lett. 21 (4) (1985) 181–185, https://doi.org/10.1016/0020-0190(85)90056-0.
[10] C. Talcott, F. Arbab, M. Yadav, Soft Agents: Exploring Soft Constraints to Model Robust Adaptive Distributed Cyber-Physical Agent Systems, Springer International Publishing, Cham, 2015, pp. 273–290.
[11] T. Kappé, B. Lion, F. Arbab, C. Talcott, Soft component automata: composition, compilation, logic, and verification, Sci. Comput. Program. 183 (2019) 102300, https://doi.org/10.1016/j.scico.2019.08.001.

[12] E.A. Lee, Cyber physical systems: design challenges, in: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), 2008, pp. 363–369.
[13] K. Kim, P.R. Kumar, Cyber–physical systems: a perspective at the centennial, Proc. IEEE 100 (2012) 1287–1308, https://doi.org/10.1109/JPROC.2012.2189792 (Special Centennial Issue).
[14] L. de Alfaro, T.A. Henzinger, Interface theories for component-based design, in: T.A. Henzinger, C.M. Kirsch (Eds.), Embedded Software, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 148–165.
[15] F. Arbab, Abstract behavior types: a foundation model for components and their composition, in: Formal Methods for Components and Objects: Pragmatic Aspects and Applications, Sci. Comput. Program. 55 (1) (2005) 3–52, https://doi.org/10.1016/j.scico.2004.05.010.
[16] J. Rutten, Universal coalgebra: a theory of systems, in: Modern Algebra, Theor. Comput. Sci. 249 (1) (2000) 3–80, https://doi.org/10.1016/S0304-3975(00)00056-6.
[17] T.A. Henzinger, The theory of hybrid automata, in: Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27–30, 1996, IEEE Computer Society, 1996, pp. 278–292.
[18] A. Arnold, Nivat's processes and their synchronization, Theor. Comput. Sci. 281 (1–2) (2002) 31–36, https://doi.org/10.1016/S0304-3975(02)00006-3.
[19] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, D. Teneketzis, Diagnosability of discrete-event systems, IEEE Trans. Autom. Control 40 (9) (1995) 1555–1575, https://doi.org/10.1109/9.412626.
[20] C. Talcott, Cyber-Physical Systems and Events, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 101–115.
[21] L. Sha, A. Al-Nayeem, M. Sun, J. Meseguer, P. Ölveczky, Pals: physically asynchronous logically synchronous systems, Tech. Rep., University of Illinois Urbana-Champaign, 06 2010.
[22] M. Kanovich1, T.B. Kirigin, V. Nigam, A. Scedrov, C. Talcott, Timed multiset rewriting and the verification of time-sensitive distributed systems, in: The 14th International Conference on Formal Modelling and Analysis of Timed Systems, vol. 9884, 2016, pp. 228–244.
[23] M. Kanovich, T.B. Kirigin, V. Nigam, A. Scedrov, C. Talcott, On the complexity of verification of time-sensitive distributed systems, in: D. Dougherty, J. Meseguer, S.A. Mödersheim, P. Rowe (Eds.), Protocols, Strands, and Logic, in: LNCS, Springer, Cham, 2021, honoring Joshua Guttman.
[24] J. Liu, J. Lv, Z. Quan, N. Zhan, H. Zhao, C. Zhou, L. Zou, A calculus for hybrid CSP, in: K. Ueda (Ed.), Programming Languages and Systems - 8th Asian Symposium, APLAS 2010, Proceedings, Shanghai, China, November 28 – December 1, 2010, in: Lecture Notes in Computer Science, vol. 6461, Springer, 2010, pp. 1–15.
[25] C.A.R. Hoare, Communicating sequential processes, Commun. ACM 21 (8) (1978) 666–677, https://doi.org/10.1145/359576.359585.
[26] X. Xu, J.-P. Talpin, S. Wang, B. Zhan, N. Zhan, Semantics foundation for cyber-physical systems using higher-order UTP, ACM Trans. Softw. Eng. Methodol. (Feb. 2022), https://doi.org/10.1145/3517192, in press.
[27] B. Finkbeiner, C. Hahn, J. Hofmann, L. Tentrup, Realizing ω-regular hyperproperties, in: S.K. Lahiri, C. Wang (Eds.), Computer Aided Verification, Springer International Publishing, Cham, 2020, pp. 40–63.
[28] L.V. Nguyen, J. Kapinski, X. Jin, J.V. Deshmukh, T.T. Johnson, Hyperproperties of real-valued signals, in: Proceedings of the 15-th ACM-IEEE International Conference on Formal Methods and Models for System Design, MEMOCODE '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 104–113.
[29] N. Lynch, R. Segala, F. Vaandrager, Hybrid i/o automata, Inf. Comput. 185 (1) (2003) 105–157, https://doi.org/10.1016/S0890-5401(03)00067-1.
[30] J.W. Ro, A. Malik, P. Roop, A compositional semantics of simulink/stateflow based on quantized state hybrid automata, in: Proceedings of the 17th ACM-IEEE International Conference on Formal Methods and Models for System Design, MEMOCODE '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 11.
[31] G. Hamon, J.M. Rushby, An operational semantics for stateflow, Int. J. Softw. Tools Technol. Transf. 9 (5–6) (2007) 447–456, https://doi.org/10.1007/s10009-007-0049-7.
[32] https://scm.cwi.nl/FM/cp-agent.
[33] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, C. Talcott, All About Maude: A High-Performance Logical Framework, Springer, 2007.