3C4-IOS-4a-2

Resilience of Event-Driven Dynamic Systems

Nicolas Schwind*1 Morgan Magnin*2 Katsumi Inoue*1

*1 National Institute of Informatics *2 École Centrale de Nantes/IRCCyN

When designing and evaluating the performances of a system in terms of resilience, it is crucial to consider not only its global configuration, but also its dynamics with regard to the environment. Indeed, our systems are subject to uncontrollable events and their analysis requires to capture these events as well as the controlled events (i.e., the actions) in their global succession relationships. In this paper, we define a language that is expressive enough to represent any *narrative scheme*, that is, any set of total preorders over any finite set of uncontrollable events. We then formalize the problem of existence of an adequate *strategy* for a given narrative scheme, that consists in adding some actions between the uncontrollable events (leading to a specific *scenario*) in a way that the system satisfies some expected property. Then, each scenario is interpreted as a sequence of propositional formulae that are updated each time an uncontrollable event or an action occurs. Lastly, we introduce some properties in order to characterize the notion of resilience for such event-driven dynamic systems.

1. Introduction

1.1 Context

After the Great East Japan Earthquake of March 11, 2011, a large number of scientists in Japan have chosen to dedicate their efforts to the design of innovative researches able to face upstream and downstream such destructive events. But this is not a concern with Japan only. The disasters and threatens of the last two decades (climate changes, 9/11 attacks, swine flu, ...) have emphasized the need for frameworks adapted to the modeling of large-scale damaging events. More than just modeling, it is a matter of being able to assess a wide range of properties that check the capabilities of the systems in terms of resistance, robustness and recovery. In other words, the research in the field of *resilient systems* and *resilient properties* is to become a hot-topic at a worldwide scale.

The analysis towards resilient properties implies to consider the general dynamics of the targeted systems. To capture their general behavior, it then becomes crucial to benefit from a concise definition of all possible dynamics and to design an elegant yet relevant way to modify these dynamics according a set of properties. That is why we decided to focus on a logical based approach.

1.2 Logical representation of discrete-event systems

First-order logics have already been revealed as very useful for the analysis of large-scale, dynamics systems. In [18], Péli and Masuch defined a fragment of organizational ecology in first-order logic. Their approach takes its roots in the observation that most theories in social sciences are generally formulated in natural language, which leads to many ambiguity and confusion, despite its expressiveness. We feel the same need to state dynamics of a system and its resilience in a formal framework suitable to the application of mathematical techniques.

Contact: Nicolas Schwind, National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan, schwind@nii.ac.jp

In [16], Naylor proposed a first methodology based on first-order logic for modeling discrete-event systems. Practically, he did not only address models with discrete events, but also considered the elapsing of time (with time being assimilated as a quantity evolving over $\mathbb{R}$). He intendedly focused on a representation of time through first-order logics, and not through any higher-order logics. This raises some practical issues he discussed. The results of these works gave the basics for discrete-event systems to be modeled through a first-order language, including variable and constant symbols, predicate and function symbols, and logic symbols. Naylor concluded by emphasizing the flexibility of this approach, yet recognizing it suffers from its complexity, making most models to be intractable. In addition, his framework does not distinguish controllable and uncontrollable events, needed for building a control theory. That is why, in the following contribution, we will consider both actions and events, the first being controllable, the second uncontrollable. This means that the latter ones can be prevented from occurring, when desired.

This work is also connected to the modeling of discrete-event systems *via* rule-based models. Rule-based modeling has been an emerging approach for fifteen years, not only in the field of discrete-event systems but also for biochemical (more generally biological) systems. In [9], the authors consider a rule-based formalism for modeling discrete-events systems with faults. Contrary to classical models of discrete-events systems, their model is of polynomial size in the number of signals and faults. They even extend their framework to incorporate delay faults, thus reasoning not only on the chronology of events, but also on their chronological succession (incorporating quantitive timing information), which implies to have more rules.

1.3 Resilience

Some recent work have considered resilience under the prism of logics. But these frameworks are generally more complex than the one we aim to define here. For instance, in [4], Bursztein and Goubault-Larrecq consider a variant of modal logics to assess the resilience of computer networks

to random faults. The originality of their approach lies in the fact that they encompass time into their model (considering then timed game automata), thus define a variant of the TATL logic [7] dedicated to the kind of properties useful for resilience. But then model-checking of TATL logic suffer from being EXPTIME-complete. Our approach is to establish a less expressive, but far more tractable framework for resilient systems evaluation.

But what exactly is resilience? The resilience concept encompasses a large range of notions, depending on the research field and the authors. This is a topic of interest for a wide set of areas. To pursue with our comparison with ecological systems, the authors of [18] define resilience as a less likely probability to die when resources are scarce.

In [13], Li et al. study resilience for the design of physical electronic circuits: they cite numerous defects that circuits must face, e.g. soft transient errors, aging and wear out, environmental variations, ... Their approach consists in giving a formal framework for a resilient control synthesis of electronic circuits. They thus define a resilience measure associated to local repairs, meaning to each individual circuit component. Meanwhile, every repair is also associated with a power *cost*. Consequently, the resilient control synthesis problem consists in an optimization problem defined as a $0-1$ integer linear problem consisting in maximizing the global resilience measure of the system while keeping the total cost below a given (power overhead) *budget*.

Finally, when coming back to the field of discrete-events systems [4], resilience is assimilated to the ability to survive attacks and faults on the one hand, recover from them on the other hand. This means resilience should encapsulate two phases in the analysis of a system: beforehand, the system is expected to resist to unexpected environment pressure; afterwards, it is expected to recover from damages at a *reasonable* cost. This is consistent with the definition of resilience in [15], where Minami et al. cover these features by the definition of resistance to disturbances on one hand, recoverability from an undesired state on the other hand.

Recently, this definition of resilience properties has been refined in [20]. The authors focus on a family of four core properties, that are:

- resistance, *i.e.* the ability for the system to absorb by itself drastic modifications of the environment;
- recoverability, *i.e.* the ability to reach an admissible state within a given time interval after an unwanted (potentially damaging) modification of the environment;
- functionality, *i.e.* the ability to guarantee an average quality of service for a given time interval;
- stabilizability, *i.e.* the ability to keep the cost for maintaining the integrity of the system under a given budget.

The existing work that is the most related to our objective here is the seminal paper of Ramadge and Wonham about the control of discrete-event systems [19]. As they aimed at designing a framework in which some key properties could be guaranteed by the addition of a controller, they needed to partition the set of events into uncontrollable and controllable events. They considered a logical representation of discrete-events systems and modeled the behavior of such a system as a prefix closed language over the event alphabet. Then they introduced the notion of supervisor, which associates every possible string of events with a control input (that is a subset of events) to be applied. The control synthesis problem is then to build a supervisor such that some undesirable sequences of events cannot occur or some desirable sequences are wished to occur.

Similar concepts to resilience, such as stabilizability [17] and maintainability [3] have also been proposed by other researchers within the structure of discrete-event systems.

1.4 Outline

With this contribution, our goal is to make a first connection between this general control synthesis problem and propositional logic so that we are able to assess the resilience of the system. In this paper, we provide a general framework for resilience properties adapted to the context of discrete-event systems. The originality of our approach lies in the logical-based modeling framework we have designed, and its relation with key resilience properties. Section 2 defines our formal context and notations. In section 3, we introduce the formal language that to captures not only exogenous events the system must endorse, but also strategies that would allow to control the system to meet specific behavior requirements. Then, in section 4, we give the set of key properties that allow us to establish the various criteria which characterize a resilient system. Section 5 summarizes the main ideas behind our formalism, discusses its advantages and gives an overview of further work.

2. Preliminaries

Let X be a finite set. $|X|$ denotes the number of elements in X. A total preorder over X (denoted $(X, \leq_X)$, or simply $\leq_X$) is a binary relation over X that is reflexive ($\forall x \in X, x \leq_X x$) and transitive ($\forall x, y, z \in X$, if $x \leq_X y$ and $y \leq_X z$, then $x \leq_X z$). Let $\leq_X$ be a total preorder. $=_X$ denotes the corresponding equivalence relation, that is, $\forall x, y$, $x =_X y$ if and only if $x \leq_X y$ and $y \leq_X x$, and $<_X$ the corresponding strict ordering, that is, $\forall x, y$, $x <_X y$ if and only if $x \leq_X y$ and not $y \leq_X x$. X can be partitioned into an ordered set of equivalent classes $EC(X) = \{X_1, X_2, \dots\}$, that is,
- $X = \bigcup\{X_i \mid X_i \in EC(X)\}$,
- $\forall X_i, X_j \in EC(X), i \neq j, X_i \cap X_j = \emptyset$,
- $\forall X_i \in EC(X), \forall x, y \in X_i, x =_X y$, and
- $\forall i, j \in \{1, \dots, |EC(X)|\}, i < j$, if $x \in X_i$ and $y \in X_j$ then $x <_X y$.

Each equivalence class X^i is also called the i^{th} class of $(X, \leq_X)$.

3. The event-based language $\mathcal{L}_E$

We consider a (limited) first-order logical setting, expressive enough for representing and reasoning about a set of possible *narratives*. Unformally, a narrative is a succession of events of environmental nature (i.e., *uncontrollable events*), that is, a narrative is characterized by a total preorder over a given set E of uncontrollable events.

3.1 Syntax and semantics of $\mathcal{L}_E$

The syntax and the semantics of our representation language, denoted $\mathcal{L}_E$, are defined as follows. The alphabet of $\mathcal{L}_E$ consists of a finite set of variables $E = \{e_1, \ldots, e_n\}$ (where each $e_i \in E$ represents an uncontrollable event), a unary predicate symbol $\preceq$, the usual logical connectives $\neg$ (not), $\wedge$ (and), $\vee$ (or), the usual constant symbols $\top$ (true) and $\bot$ (false), and the punctuation symbols '(' and ')'. An *atom* is of the form $(e_i \preceq e_j)$, with $e_i, e_j \in E$. The language $\mathcal{L}_E$ is inductively defined as follows: every atom is a formula, $\top$ and $\bot$ are formulae and given two formulae α and β, $\neg\alpha$, $\alpha \wedge \beta$ and $\alpha \vee \beta$ are formulae.

We now define the notion of *narrative*:

Definition 1 (Narrative) *A* narrative ω *(over E) is a total preorder $(E, \leq_E)$.*

The semantics of an atom of the form $(e_i \preceq e_j)$ for a given narrative $\omega = (E, \leq_E)$ is defined as $[\![(e_i \preceq e_j)]\!] = true$ if and only if $e_i \leq_E e_j$. The set of all possible narratives is denoted Ω. A narrative is a *model* of a formula α (denoted $\omega \models \alpha$) if and only if it makes the formula α true in the usual truth functional way. The set of all models of a formula α is denoted $mod(\alpha)$. In the following, a formula from $\mathcal{L}_E$ is also called a *narrative scheme*:

Definition 2 (Narrative scheme) *A* narrative scheme *is a formula from $\mathcal{L}_E$.*

Example 1 *Assume that two earthquakes e_1, e_2 are expected to occur independently. It is expected that e_1 will be followed by a tsunami e_3 and that e_2 will be be followed by a tsunami e_4. No other event is expected. This information is represented by the narrative scheme $\alpha = (e_1 \preceq e_3) \wedge \neg(e_3 \preceq e_1) \wedge (e_2 \preceq e_4) \wedge \neg(e_4 \preceq e_2)$ over $E = \{e_1, e_2, e_3, e_4\}$.*

The example above shows that one can represent a (possibly exponential-sized) set of narratives using a simple formula from our representation language $\mathcal{L}_E$. Moreover, $\mathcal{L}_E$ is expressive enough to represent *any* possible set of narratives over E, as it is stated in the following proposition:

Proposition 1 *For any set S of narratives over E, there exists a narrative scheme α such that $mod(\alpha) = S$.*

Proof: Let $S = \{\omega_1, \ldots, \omega_m\}$ be a set of narratives over E. For every $\omega_k \in S$ with $\omega_k = (E, \leq_E^k)$, let α_k the formula form $\mathcal{L}_E$ defined as

$$\alpha_k = \bigwedge\{(e_i \preceq e_j) \mid e_i, e_j \in E, e_i \leq_E^k e_j\}.$$

We have $mod(\alpha_k) = \omega_k$. Now, let α be the formula from $\mathcal{L}_E$ defined as

$$\alpha = \bigvee\{\alpha_k \mid k \in \{1, \ldots, m\}\}.$$

We have $mod(\alpha) = S$. ■

Given a narrative scheme α, our motivation comes from the "control" of the narratives from $mod(\alpha)$ so that they satisfy some expected property. More precisely, we are given a finite set of *actions* A (i.e., controllable events) that can be inserted *between* the uncontrollable events from any given narrative. A narrative completed by some actions is called a *scenario*:

Definition 3 (Scenario) *A* scenario σ *is a total preorder over $E \cup A_p$ where $A_p \subseteq A$.*

Given a finite set of actions A, we denote $scen(\omega)$ the set of all possible scenarios that "complete" the narrative ω. Formally, for every narrative $\omega = (E, \leq_E)$,

$$scen(\omega) = \{(E \cup A_p, \leq_{E \cup A_p}) \mid A_p \subseteq A, \forall e_i, e_j \in E, \\ e_i \leq_E e_j \iff e_i \leq_{E \cup A_p} e_j, \forall e_i \in E, \forall a_i \in A_p, a_i \neq e_i\}.$$

Example 1 (continued) *Consider again the narrative scheme α from our running example. The narrative $\omega = e_1 <_E e_2 =_E e_3 <_E e_4$ is a model of α. Consider the set $A = \{a_1, a_2, a_3\}$ of actions. Then the total preorder $\sigma = e_1 <_{E \cup A_p} a_1 <_{E \cup A_p} e_2 =_{E \cup A_p} e_3 <_{E \cup A_p} a_2 <_{E \cup A_p} e_4$ over $E \cup A_p$ where $A_p = \{a_1, a_2\}$, represents a scenario resulting from a "strategy" applied on ω. That is, we have $\sigma \in scen(\omega)$.*

We formally define the notion of *strategy* within a set of actions A:

Definition 4 (Strategy) *Given a set A of actions, a* strategy within A *is a mapping strat that associates every narrative $\omega \in \Omega$ with a scenario from $scen(\omega)$.*

Not every strategy can be realized in such event-driven dynamic systems. Indeed, some of them should be discarded, as it is shown in the following example:

Example 1 (continued) *From our running example, consider again the narrative $\omega = e_1 <_E e_2 =_E e_3 <_E e_4$ that is a model of the narrative scheme α, and the additional narrative $\omega' = e_1 <_E e_2 <_E e_3 =_E e_4$ that is also a model of α. Consider the strategy within A that associates with the narratives ω and ω' the scenarios over $E \cup A'_p$ with $A'_p = \{a_1\}$ defined as follows:*

$$\begin{cases} strat(\omega) = e_1 <_{E \cup A'_p} a_1 <_{E \cup A'_p} e_2 =_{E \cup A'_p} e_3 <_{E \cup A'_p} e_4, \\ strat(\omega') = e_1 <_{E \cup A'_p} e_2 <_{E \cup A'_p} a_1 <_{E \cup A'_p} e_3 =_{E \cup A'_p} e_4. \end{cases}$$

This strategy cannot be realized. Indeed, for the narrative ω, strat considers to apply the action a_1 just after the occurence of the event e_3, and a_1 should then be followed by the specific sequence $e_2 =_{E \cup A'_p} e_3 <_{E \cup A'_p} e_4$ of uncontrollable events (i.e., after the action a_1, the uncontrollable

events e_2 and e_3 occur at the same time). On the other hand, for the narrative ω', strat considers to apply the action a_1 after the occurence of the uncontrollable event e_2, that does not occur anymore at the same time as the uncontrollable event e_3.

The problem here is that, in the practical case, after the occurrence of the uncontrollable event e_3, one should not be able to guess which narrative from $\{\omega, \omega'\}$ will actually correspond to the factual run, while here, applying the action a_1 just after e_1 discards the narrative ω' from a possible future situation. This is an undesirable behavior for a strategy, as the actions are only supposed to prevent *the system from an undesirable state or* repair *it, while they are not supposed to rule the ordering of the uncontrolled events occuring next to these actions. An example of a well-behaved strategy $strat'$ with respect to the narratives ω, ω' is defined as follows:*

$$\begin{cases} strat'(\omega) = e_1 <_{E\cup A'_p} e_2 =_{E\cup A'_p} e_3 <_{E\cup A'_p} a_1 <_{E\cup A'_p} e_4, \\ strat'(\omega') = e_1 <_{E\cup A'_p} e_2 <_{E\cup A'_p} a_1 <_{E\cup A'_p} e_3 =_{E\cup A'_p} e_4. \end{cases}$$

Before we formally characterize which are the well-behaved strategies (i.e., the *realizable* ones), we introduce some preliminary notions.

Definition 5 (Strong completion) *Given two total preorders $(S, \leq_S)$ and $(S', \leq_{S'})$, we say that $(S', \leq_{S'})$* strongly completes *$(S, \leq_S)$ if $S \subseteq S'$ and for every $x, y \in S$, $x \leq_S y$ if and only if $x \leq_{S'} y$.*

Example 1 (continued) *The scenario $\sigma = e_1 <_{E\cup A'_p} e_2 =_{E\cup A'_p} e_3 <_{E\cup A'_p} a_1 <_{E\cup A'_p} e_4$ strongly completes the narrative $\omega = e_1 <_E e_2 =_E e_3 <_E e_4$.*

Definition 6 (Weak completion) *Given two total preorders $(S, \leq_S)$ and $(S', \leq_{S'})$, we say that $(S', \leq_{S'})$* weakly completes *$(S, \leq_S)$ if $(S', \leq_{S'})$ strongly completes $(S \cap S', \leq_{S\cap S'})$, where $(S \cap S', \leq_{S\cap S'})$ is strongly completed by $(S, \leq_S)$.*

Example 1 (continued) *The narrative $\omega = e_1 <_E e_2 =_E e_3 <_E e_4$ weakly completes the scenario $\omega = e_1 <_{E\cup A'_p} e_2 =_{E\cup A'_p} e_3 <_{E\cup A'_p} a_1$.*

Definition 7 (Partial run) *Given two total preorders $(S, \leq_S)$ and $(S', \leq_{S'})$, $(S, \leq_S)$ is said to be a* partial run *of $(S', \leq_{S'})$ if $(S', \leq_{S'})$ strongly completes $(S, \leq_S)$ and for every $x \in S'$ and every $y \in S \setminus S'$, we have $x <_S y$.*
The set of all partial runs of a given total preorder $(S, \leq_S)$ is denoted $Partials((S, \leq_S))$.

Example 1 (continued) *$\sigma_p = e_1 <_{E\cup A'_p} e_2 =_{E\cup A'_p} e_3$ is a partial run of the scenario $\sigma = e_1 <_{E\cup A'_p} e_2 =_{E\cup A'_p} e_3 <_{E\cup A'_p} a_1 <_{E\cup A'_p} e_4$.*

We are now ready to define the notion of *realizable* strategy.

Definition 8 (Realizable strategy) *A strategy strat is said to be* realizable *if for every partial run $\sigma_p \in \bigcup\{Partials(strat(\omega)) \mid \omega \in \Omega\}$, for every narrative $\omega \in \Omega$ that weakly completes σ_p, the scenario $strat(\omega)$ strongly completes σ_p.*

Intuitively, for every partial run σ_p induced from a strategy, every narrative that completes it (i.e., every possible future that has begun as described by σ_p) should be also considered in the strategy. That is, for each narrative ω that weakly completes σ_p there must exist a scenario induced from the strategy that begins in the same way as described by σ_p and that strongly completes ω. Therefore, in the following we shall restrict ourselves to realizable strategies.

We are then interested in the following (generic) problem:

Problem 1 *Given a narrative scheme α over E, a set of actions A and a property P, does there exist a realizable strategy strat within A such that each scenario from $\bigcup\{strat(\omega) \mid \omega \models \alpha\}$ satisfies P?*

The next section defines a simple class of event-driven dynamic systems that take advantage of the narrative schemes from $\mathcal{L}_E$ defined in the previous section. In addition, we will introduce some specific properties related to the resilience of such event-driven dynamic systems with the objective of concretizing the above problem.

4. Resilience of event-based dynamic systems using propositional logic

4.1 Dynamic systems

This section introduces a formalism that defines an event-driven dynamic system in a simple way. In addition to the language $\mathcal{L}_E$, we consider here a propositional language $\mathcal{L}_{PROP}$ defined from a finite set of propositional variables $PROP$ that represent the entities which specify the system (i.e., the components of the system). An interpretation is a mapping from $PROP$ to $\{0, 1\}$. The set of all interpretations is denoted $\mathcal{W}$. An interpretation I is a model of a formula $\phi \in \mathcal{L}_{PROP}$, denoted $I \models \phi$ if and only if it makes it true in the usual truth functional way. $mod(\phi)$ denotes the set of models of formula $\phi \in \mathcal{L}_{PROP}$. A formula from $\mathcal{L}_{PROP}$ is said to be *consistent* if there exists a model of this formula. Two formulae from ϕ, ϕ' from $\mathcal{L}_{PROP}$ are said to be *equivalent*, denoted $\phi \equiv \phi'$ if $mod(\phi) = mod(\phi')$.

Definition 9 (Dynamic System) *A* dynamic system *DS is a tuple $\langle \phi_0, intcost, \alpha, A, actioncost, f, \diamond, dist \rangle$, where:*

- *ϕ_0 is a formula from $\mathcal{L}_{PROP}$ representing the initial system specifications of DS;*
- *$intcost$ is an* interpretation cost function*, that is, a mapping from A to $\mathbb{R}^+$ that represents the cost of each interpretation from $\mathcal{W}$;*
- *α is a narrative scheme from $\mathcal{L}_E$;*
- *A is a finite set of actions;*
- *$actioncost$ is an* action cost function*, that is, a mapping from A to $\mathbb{R}^+$ that represents the cost of each action from A; for simplicity in the rest, we consider that $actioncost$ is a mapping from $E \cup A$ to $\mathbb{R}^+$ such that for every $e_i \in E$, $actioncost(e_i) = 0$;*

- *f is a mapping associating every uncontrollable event and every action from $E \cup A$ with a formula from $\mathcal{L}_{PROP}$;*
- *$\diamond$ is an (update) operator that associates every pair of formulae ϕ, ϕ' from $\mathcal{L}_{PROP}$ with a propositional formulae denoted $\phi \diamond \phi'$ which represents the* update *of ϕ by ϕ';*
- *dist is a* transitional cost *between interpretations, that is a premetric, i.e., dist is a mapping from $\mathcal{W} \times \mathcal{W}$ to $\mathbb{R}^+$ such that for every interpretation I, $dist(I, I) = 0$.*

From now on, we are given a dynamic system DS. DS associates with every scenario from $\bigcup\{scen(\omega) \mid \omega \models \alpha\}$ a path of systems specifications, also called *systems path* (defined below). Each system of a given systems path is associated with a formula from $\mathcal{L}_{PROP}$ describing the constraints inherent to the system at a given time within the scenario. Then, for a given formula ϕ from $\mathcal{L}_{PROP}$ associated with a system from a systems path, each model I of ϕ represents a specific *configuration* of the system specified by ϕ. That is to say, such a model I can also be viewed as a specific *state* of the system at a given time within a given scenario. Moreover, I is associated with a cost specified by $intcost(I)$ that allows us to evaluate the quality of the state represented by I. That is, *intcost* provides a way to discriminate the states of a given system from each other in terms of "quality". The use of the transitional cost function *dist* will be explained later in the paper.

Given a scenario σ, let us denote σ_i the i^{th} class of σ [*1]:

Definition 10 (Systems path) *Let σ be a scenario from $\bigcup\{scen(\omega) \mid \omega \models \alpha\}$. The* systems path *associated with σ, denoted $SP(\sigma)$ is the sequence $(\phi_0, \phi_1, \dots)$ of formulae from $\mathcal{L}_{PROP}$ such that for every $\phi_i \in SP(\sigma)$, $i > 0$,*

$$\phi_i = \begin{cases} \phi_{i-1} \diamond \bigwedge\{f(x) \mid x \in \sigma_i\} & \text{if } \sigma_i \subseteq E, \\ \phi_{i-1} \diamond \bigvee\{f(x) \mid x \in \sigma_i\} & \text{otherwise (if } \sigma_i \subseteq A). \end{cases}$$

For the sake of clarity, we also write a systems path $SP(\sigma)$ as follows:

$$(\phi_0 \xrightarrow{\sigma_1} \phi_1 \xrightarrow{\sigma_2} \phi_2 \xrightarrow{\sigma_3} \dots).$$

Intuitively, given a specific scenario σ we build the associated systems path $(\phi_0, \phi_1, \dots)$ as follows: we consider ϕ_0 as the initial systems specifications of the dynamic system, and then we simulate a "run" of σ. We encounter first either a set of uncontrollable events, or a set of actions. In the former case, we define the formula from $\mathcal{L}_{PROP}$ describing the next system ϕ_1 as the update of ϕ_0 by the *conjunction* of all formulae associated (by f) with the encountered uncontrollable events; in the latter case, the next system ϕ_1 is defined as the update of ϕ_0 by the *disjunction* of all formulae associated (by f) with the encountered actions. Indeed, it is natural to consider that a set of uncontrollable events are interpreted conjunctively since the constraints they represent should be considered together. On the other hand, a set of actions are interpreted disjunctively since they should be designed to relax the constraints of the system.

We now focus on the behaviour that should be adopted by the operator $\diamond$ that is used to *update* the previous systems specifications by either the conjunction of formulae associated with the encountered events (in case of uncontrollable events), or the disjunction of them (in case of controllable ones). *Updating* a propositional formula by an other propositional formula has been topic that has been a widely studied topic these past 25 years [1, 12, 10, 11]. The operation of update consists in bringing the systems specifications up-to-date when the world described by them changes. That is a convenient type of operation in our framework, since we consider that the systems specifications evolve by the occurence of some events [*2]. The expected behaviour for update operators is captured by the following set of properties, usually referred as the *KM postulates* in the literature [10, 11]:

Definition 11 (Update operator) *An operator $\diamond$ that associates every propositional formula ϕ, ϕ' with a propositional formula $\phi \diamond \phi'$ is an* update operator *if and only if for every formula $\phi, \phi_1, \phi_2, \phi', \phi'_1, \phi'_2$, it satisfies the following postulates:*

(U1) $\phi \diamond \phi' \models \phi'$;

(U2) *If $\phi \models \phi'$, then $\phi \diamond \phi' \equiv \phi$;*

(U3) *If ϕ is consistent and ϕ' is consistent, then $\phi \diamond \phi'$ is consistent;*

(U4) *If $\phi_1 \equiv \phi_2$ and $\phi'_1 \equiv \phi'_2$, then $\phi_1 \diamond \phi'_1 \equiv \phi_2 \diamond \phi'_2$;*

(U5) $(\phi \diamond \phi'_1) \wedge \phi'_2 \models \phi \diamond (\phi'_1 \wedge \phi'_2)$;

(U6) *If $(\phi \diamond \phi'_1) \models \phi'_2$ and $(\phi \diamond \phi'_2) \models \phi'_1$, then $\phi \diamond \phi'_1 \equiv \phi \diamond \phi'_2$;*

(U7) *If $mod(\phi) = 1$, then $(\phi \diamond \phi'_1) \wedge (\phi \diamond \phi'_2) \models \phi \diamond (\phi'_1 \vee \phi'_2)$;*

(U8) $(\phi_1 \vee \phi_2) \diamond \phi' \equiv (\phi_1 \diamond \phi') \vee (\phi_2 \diamond \phi')$.

For instance, the postulate **(U1)** requires that the models of the propositional formula resulting from the update of ϕ by ϕ' should also be models of ϕ' (that is, a postulate of *success*). The postulate **(U8)** expresses the fact that updating a propositional formula by an other one is made in a model-wise fashion. We refer the reader to [10, 11] for more details about the rationale of these postulates.

Update operators include the drastic update operator $\diamond_D$ which is defined for every propositional formula ϕ, ϕ' as:

$$\phi \diamond_D \phi' \equiv \begin{cases} \phi & \text{if } \phi \models \phi', \\ \phi' & \text{otherwise.} \end{cases}$$

*1 Please note that each equivalent class of a scenario is composed either of uncontrollable events, or of actions only.

*2 In comparison, *revising* a propositional formula by an other one [2] is more appropriated in the context where we are obtaining new information about a static world, i.e., when the new information does not describe a *change* by the occurence of an action or uncontrollable event, but instead represents some information that describes the same, static system, and that is more *accurate*.

More parsimonious update operators can be considered, for instance, the Forbus operator $\diamond_F$ [5] which is defined as follows: let d_H the Hamming distance between interpretations, that is, the number of atom values that differ between two given interpretations. Then $\diamond_F$ is defined for every propositional formula ϕ, ϕ', as

$$\phi \diamond_F \phi' \equiv \bigvee_{\omega' \models \phi} \{\omega \models \phi' \mid d_H(\omega, \omega') \text{ is minimal}\}.$$

Many such update operators can be defined, and for our dynamic systems any update operator $\diamond$ can be used, as far as it is an update operator in the sense of Definition 11.

We now introduce the definition of *models path*, that depends on a specific systems path:

Definition 12 (Models path) *Given a systems path $sp = (\phi_0, \phi_1, \dots)$, a* models path *of sp is a sequence $(I_0, I_1, \dots)$ such that for every $k \in \{0, 1, \dots\}$, $I_k \models \phi_k$. A* models sub-path *sp' of sp is a subsequence $(I_a, \dots, I_b)$ of sp, with $a \leq b$.*

We recall that each interpretation I_k from a models path of a given systems path represents a specific configuration of the systems specifications given by ϕ_k, that is, $I_k \models \phi_k$. Stated otherwise, each interpretation from a models path represents a specific state of the system at a given time within a given scenario σ, where $SP(\sigma)$ is the systems path under consideration.

4.2 Resilience of event-based dynamic systems

We are now ready to introduce the properties of interest with respect to resilience into our framework.

4.2.1 Consistency

We propose a first property that we believe is mandatory for dynamic systems, that is, the property of *consistency* of a dynamic system. This property is introduced on systems path first, and will be extended to dynamic systems straightforwardly. Intuitively, a systems path is *consistent* if it is always possible to define the state of the system all allong the systems path, that is, if one can always configure the system all along its life in a given scenario. This property can be viewed as the counterpart in our framework of a widely accepted key feature of resilience, that is, that it should always be possible to maintain the system's core purpose and integrity in the face of dramatically changed circumstances.

Definition 13 (Consistent systems path / scenario) *A systems path $sp = (\phi_0, \phi_1, \dots)$ is said to be* consistent *if every formula appearing in sp is consistent.*
A scenario σ is said to be consistent *if $SP(\sigma)$ is a consistent systems path.*

Example 1 (continued) *Consider again the scenario from our running example, that is the scenario $\sigma = e_1 <_{E \cup A_p} a_1 <_{E \cup A_p} e_2 =_{E \cup A_p} e_3 <_{E \cup A_p} a_2 <_{E \cup A_p} e_4$ over $E \cup A_p$ where $A_p = \{a_1, a_2\}$. Let $PROP = \{a, b\}$, $\phi_0 = a \vee b$, and f be defined as follows:*

$$f(e_1) = a \wedge b, \quad f(e_2) = \neg a, \quad f(e_3) = a,$$
$$f(e_4) = a \wedge \neg b, \quad f(a_1) = b, \quad f(a_2) = a \vee b.$$

Assume that the update operator under consideration in DS is the drastic operator $\diamond_D$. Then the systems path $SP(\sigma)$ is defined as the following sequence of formulae from $\mathcal{L}_{PROP}$:

$$(a \vee b \xrightarrow{e_1} a \wedge b \xrightarrow{a_1} \neg a \vee \neg b \xrightarrow{e_2, e_3} \bot \xrightarrow{a_2} \bot \xrightarrow{e_4} \bot).$$

The above example puts in light a scenario where the dynamic system faces dramatically changed circumstances that correspond to the occurence of events e_2 and e_3 simultaneously. One can see that whatever happens after the simultaneous occurence of e_2 and e_3, since $f(e_2) \wedge f(e_3)$ is an inconsistent formula, the next system in the systems path is necessarily associated with an inconsistent formula. Indeed, the postulate **(U1)** required to be satisfied by any update operator $\diamond$ demands that the models of the propositional formula resulting from the update of any formula by an inconsistent one should be inconsistent. By recurrence, all the following systems in the systems path are necessarily associated with inconsistent formulae. This kind of situation represents the case where, for instance, several disasters occur at the same time and the dynamic system is not able to recover from the damages (that is to say, one cannot configure the system anymore in the remaining part of the studied scenario), Moreoever, it can be easily seen that the consistency of a scenario (w.r.t. Definition 13) is independent from the actions that are inserted into the underlying narrative. That is, a scenario σ is inconsistent if and only if the narrative that it completes (i.e., the narrative ω such that $\sigma \in scen(\omega)$) is also inconsistent. We are now ready to extend the property of consistency to dynamic systems:

Definition 14 (Consistent dynamic system) *A dynamic system $DS = \langle \phi_0, intcost, \alpha, A, actioncost, f, \diamond, dist \rangle$ is said to be* consistent *if ϕ_0 is consistent and each narrative from $mod(\alpha)$ is consistent.*

We introduce now some additional properties that we consider as relevant for the characterization of resilient dynamic systems. They are somewhat similar to the properties proposed in [20], some of them being actually their direct counterpart. These properties are first introduced on models paths, and then will be naturally extended to systems paths and dynamic systems.

4.2.2 Resistance

Resistance is the ability for a dynamic system to absorb by itself drastic modifications of the environment.

Definition 15 (Resistance) *Given a non-negative real number l, a models path $mp = (I_0, I_1, \dots)$ is said to be* l-resistant *if for each $k \in \{0, 1, \dots\}$, $intcost(I_k) \leq l$.*

Intuitively, a models path is l-resistant if the cost of each one of its *configurations* is kept under the threshold l.

4.2.3 Recoverability

Recoverability is the ability for a dynamic system to reach an admissible state within a given total amount of action cost after an unwanted (potentially damaging) modification of the environment.

Definition 16 (Recoverability) *Le p, c be two non-negative real numbers, and let $mp = (I_0, I_1, \dots)$ be a models path of a systems path $sp = (\phi_0, \phi_1, \dots)$ associated with a scenario σ. mp is said to be* $\langle p, c\rangle$-recoverable *if for each one of its models sub-paths $(I_a, \dots, I_b)$ such that for every $k \in \{a, \dots, b\}$, $intcost(I_k) > p$, we have $\sum_{k=a}^{b} actioncost(\sigma_{a+1}) \leq c$.*

Intuitively, c represents the total amount of cost (i.e., the sum of costs of each action applied between systems specifications of sp for which the configurations in mp have a cost above p) that is allowed for a $\langle p, c\rangle$-recoverable models path to get back to a "safe" state (that is, to get back to a configuration whose cost is under p).

This notion of recoverability slightly differs from the one proposed in [20]. Indeed, in [20] the authors proposed a notion of $\langle p, q\rangle$*-recoverability*, where p, q are both non-negative real numbers and q represents the total amount of extra cost of the *interpretations* (i.e., costs above p) that is allowed for a $\langle p, q\rangle$-recoverable models path to get back to a "safe" state. This cumulative extra cost is different from ours in the sense that in our definition, we consider the total amount of *action* costs. This choice is motivated by the fact that in [20], the dynamics of the system is driven by *time* which is supposed to be regular, while in our framework the dynamics of the system is *event-driven*; in consequence, since the time between events is not assumed here to be regular, summing the extra costs of each interpretation in a models path would have no more sense.

4.2.4 Stabilizability

Stabilizability is the ability for a dynamic system to avoid undergoing changes that are associated with high transitional costs. Here, we use the premetric *dist* specified in the dynamic system. *dist* is used to represent a transitional cost function over interpretations that stands for passing from a state to an other one. This transitional cost is of different nature compared to the cost $intcost(I)$ associated with every interpretation I (see [20] for more details about the notion of transitional cost between interpretations). For instance, *dist* can be defined as the Hamming distance between interpretations (that is, the number of atom values that differ between two given interpretations), as it is an appropriate choice, e.g., in the context of dynamic SAT problems with decision change costs [8, 6].

Definition 17 (Stabilizability) *Given a non-negative real number s, a models path $mp = (I_0, I_1, \dots)$ is said to be* s-stabilizable *if for each $k \in \{1, 2, \dots\}$, $dist(I_{k-1}, I_k) \leq s$.*

4.2.5 Resilience

We are now ready to extend the properties of resistance, recoverability and stabilizability to systems paths, scenarios and dynamic systems.

Definition 18 (Resilient systems path / scenario) *Let l, p, c, s be four non-negative real numbers. A systems path is said to be* $\langle l, p, c, s\rangle$-resilient *if it admits a models path that is l-resistant, $\langle p, c\rangle$-recoverable and s-stabilizable. A scenario is said to be* $\langle l, p, c, s\rangle$-resilient *if $SP(\sigma)$ is a $\langle l, p, c, s\rangle$-resilient systems path.*

Definition 19 (Resilient dynamic system) *Let l, p, c, s be four non-negative real numbers. A dynamic system $DS = \langle \phi_0, intcost, \alpha, A, actioncost, f, \diamond, dist\rangle$ is said to be* $\langle l, p, c, s\rangle$-resilient *if there exists a realizable strategy within A such that each scenario from $\bigcup\{strat(\omega) \mid \omega \models \alpha\}$ is $\langle l, p, c, s\rangle$-resilient.*

We can now express in a simple way the main problem of interest in this paper (presented in a generic way in Section 3.):

Problem 2 *Given a dynamic system DS and four non-negative real number l, p, c, s, is DS $\langle l, p, c, s\rangle$-resilient?*

This (decision) problem is expected to be computationally hard to solve in the general case. This challenging question will be investigated as a further work.

5. Conclusion

In order to design systems that are able to face drastic destroying environment changes, the study of event-driven dynamic systems and their associated resilience properties is of primary importance. This challenging modeling issue must consider multiple-scale systems, inducing a potentially intractable complexity. That is why, in this paper, we have defined a concise logical-based approach to describe systems behaviors and their environment. We have first introduced an event-based language allowing to represent the dynamic modifications of the system and its environment. In line with the control theory framework of discrete-event systems, we have encompassed the distinction between uncontrollable and controllable events (the latter ones being called actions) in the core of our formalism. We have also integrated the notions of quality of a system and cost to perform some controllable actions: the first one reports the core concept of quality of service, this means the key factor that may be damaged (or repaired) by events (or actions); the second one illustrates the natural concept that some actions may be more expensive (or cheaper) than other ones. We have discussed the integration of actions, events and updating features in the context of dynamic systems, then defined the associated resilience properties. These properties, based on resistance, recoverability and stabilizability, are consistent with the state-of-the-art as summarized in the introduction. They cover not only the general robustness of the system, but also its qualitative reactivity. In this paper, we have given a taste about how our framework behaves in the context of various toy examples. Further work now consist in applying this framework to a real-life case-study and illustrating the merits of this formalization to analyze large-scale critical scenarios. We also plan to provide a timed extension of this methodology in order to include quantitative timing information. Time will be indeed needed to depict additional properties associated to resilience, like the notion of window of vulnerability [14].

Acknowledgements

This work is supported in part by the "Systems Resilience" project of Transdisciplinary Research Integration Center.

References

[1] Serge Abiteboul and Gösta Grahne. Update semantics for incomplete databases. In *Proceedings of 11th International Conference on Very Large Data Bases (VLDB'85)*, pages 1–12, Stockholm, Sweden, August 1985.

[2] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[3] Chitta Baral, Thomas Eiter, Marcus Bjäreland, and Mutsumi Nakamura. Maintenance goals of agents in a dynamic environment: Formulation and policy construction. *Artificial Intelligence*, 172(12-13):1429–1469, August 2008.

[4] Elie Bursztein and Jean Goubault-larrecq. A logical framework for evaluating network resilience against faults and attacks. In *Proceedings of the 12th annual Asian Computing Science Conference (ASIAN'07)*, pages 212–227, Doha, Qatar, December 2007.

[5] Kenneth D. Forbus. Introducing actions into qualitative simulation. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1273–1278, Detroit, MI, USA, August 1989.

[6] Daisuke Hatano and Katsutoshi Hirayama. Dynamic SAT with decision change costs: Formalization and solutions. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 560–565, Barcelona, Spain, July 2011.

[7] Thomas A. Henzinger and Vinayak S. Prabhu. Timed alternating-time temporal logic. In *Proceedings of the 4th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'06)*, pages 1–17, Paris, France, September 2006.

[8] Holger H. Hoos and Kevin O"Neill. Stochastic local search methods for dynamic SAT - an initial investigation. Technical report, In the AAAI'2000 Workshop on Leveraging Probability and Uncertainty in Computation, Austin, Texas, USA, July 2000.

[9] Z. Huang, V. Chandra, S. Jiang, and R. Kumar. Modeling discrete event systems with faults using a rules based modeling formalism. In *Proceedings of the 41st IEEE Conference on decision and control*, pages 4012–4017, Las Vegas, NV, USA, December 2002.

[10] Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 387–394, Cambridge, MA, USA, April 1991.

[11] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1991.

[12] Arthur M. Keller and Marianne W. Wilkins. On the use of an extended relational model to handle changing incomplete information. *IEEE Transactions on Software Engineering*, 11(7):620–633, 1985.

[13] Wenchao Li, Susmit Jha, and Sanjit A. Seshia. Generating control logic for optimized soft error resilience. In *Proceedings of the 9th Workshop on Silicon Errors in Logic - System Effects (SELSE'13)*, Palo Alto, CA, USA, March 2013.

[14] Richard Lippmann, Seth Webster, and Douglas Stetson. The effect of identifying vulnerabilities and patching software on the utility of network intrusion detection. In *Proceedings of the 5th International Conference on Recent Advances in Intrusion Detection (RAID'02)*, pages 307–326, Berlin, Heidelberg, 2002. Springer-Verlag.

[15] Kazuhiro Minami, Tenda Okimoto, Tomoya Tanjo, Nicolas Schwind, Hei Chan, Katsumi Inoue, and Hiroshi Maruyama. Formalizing the resilience of open dynamic systems. In *Proceedings of the Joint Agent Workshop and Symposium (JAWS'12)*, Kakegawa, Japan, October 2012.

[16] A.W. Naylor. *First-order Logic Models for Real-time, Discrete-event Systems*. Technical report (University of Michigan. Department of Electrical Engineering and Computer Science). University of Michigan, Computer Science and Engineering Division, Department of Electrical Engineering and Computer Science, 1993.

[17] Cuneyt M. Özveren, Alan S. Willsky, and Panos J. Antsaklis. Stability and stabilizability of discrete event dynamic systems. *Journal of the ACM*, 38(3):7300–7752, 1991.

[18] Gábor Péli and Michael Masuch. The Logic of Propagation Strategies: Axiomatizing a Fragment of Organizational Ecology in First-Order Logic. *Organization Science*, 8(3):310–331, 1997.

[19] Peter J. G. Ramadge and W. Murray Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, January 1989.

[20] Nicolas Schwind, Tenda Okimoto, Katsumi Inoue, Hei Chan, Tony Ribeiro, Kazuhiro Minami, and Hiroshi Maruyama. Systems resilience: A challenge problem for dynamic constraint-based agent systems. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13) (to appear)*, Saint Paul, MN, USA, May 2013.