Planning and Execution with Flexible Timelines: a Formal Account

Marta Cialdea Mayer · Andrea Orlandini ·
Alessandro Umbrico

Abstract Planning for real world problems with explicit temporal constraints is a challenging problem. Among several approaches, the use of flexible timelines in Planning and Scheduling has been shown to be successful in a number of concrete applications, such as, for instance, autonomous space systems.

This paper builds on previous work and presents a revised and extended formal account of flexible timelines with the aim of providing a general semantics for related planning concepts such as domains, goals, problems, constraints, and flexible plans. Some sources of uncertainty are also modeled in the proposed framework and taken into account in the characterization of valid plans that are assumed not to take decisions on components the planner cannot control. A formal definition of different forms of plan controllability is also proposed.

1 Introduction

Planning for real world problems with explicit temporal constraints is a challenging problem [6]. Among different approaches, the use of flexible timelines in Planning and Scheduling (P&S) has been shown to be successful in a number of concrete applications, such as, for instance, autonomous space systems [9, 24, 28]. Timeline-based planning has been introduced by Muscettola [28], under a modeling assumption inspired by classical control theory. A planning problem is modeled by identifying a set of relevant components whose temporal evolution must be controlled to obtain a desired behavior. Components represent logical or physical subsystems whose state may vary over time. The behavior of the domain features under control are modeled as temporal functions whose values have to be decided

M. Cialdea Mayer · A. Umbrico
Università degli Studi Roma Tre, Rome, Italy
E-mail: cialdea@dia.uniroma3.it, alessandro.umbrico@uniroma3.it

A. Orlandini
CNR-ISTC, National Research Council, Rome, Italy
E-mail: andrea.orlandini@istc.cnr.it

over a temporal horizon. Such functions are synthesized during problem solving by posting planning decisions. The evolution of a single temporal feature over a temporal horizon is called the *timeline* of that feature.

In general, plans synthesized by temporal P&S systems may be (i) temporally flexible and (ii) not fully controllable. Time flexibility reflects on modeling plans as made up of *flexible* timelines, describing transition events that are associated with temporal intervals (with given lower and upper bounds), instead of exact temporal occurrences. In other words, a flexible plan describes an envelope of possible solutions with the aim of facing uncertainty during actual execution. As a matter of fact, many P&S architectures return flexible plans, which are commonly accepted to be less brittle than fully specified plans, when coping with execution. The second above-mentioned property is due to the fact that not every value transition in a plan is under the system control, as events exist that depend on the *environment*. The execution of a flexible plan is usually under the responsibility of an executive system that forces value transitions over the timelines dispatching commands to the concrete system, while continuously accepting feedback and, thus, monitoring plan execution. In such cases, the execution time of controllable tasks should be chosen so that they can face uncontrollable events. This is known as the *controllability problem* [35].

A remarkable research effort has been dedicated to design, build, and deploy software environments for the synthesis of timeline-based P&S applications, including EUROPA [3], ASPEN [13] and APSI-TRF [12] and, mostly related to them, several attempts have also been made to characterize the concept of timeline. Cesta and Oddi [8] describe a domain description language able to represent physical domains to solve P&S problems. Frank and Jónsson [20] present a constraint-based attribute and interval planning paradigm for representing and reasoning about temporal plans. A characterization of the paradigm, based on temporal intervals and attributes is given in order to enable the description of planning domains with time, resources, concurrent activities, etc. Chien et al. [14] describe a basic timeline representation to represent a set of states, resources, timing, and transition constraints, aiming at generalizing previous efforts provided in a number of P&S systems designed for space applications. Frank [19] aims at defining timelines from a knowledge engineering point of view (i.e., discussing it as a data structure with associated services that a planning system may use in order to solve problems).

The above cited works aim to describe the exploited concepts, languages and tools, but do not provide a formally grounded definition for them. A first comprehensive and semantically well-founded framework for timeline-based planning is provided by Cimatti et al. [17], including temporal uncertainty but without explicitly considering time flexibility. In Bernardini's PhD thesis work [5], temporal flexibility is also considered but temporal networks are exploited in order to represent constraints. Moreover, controllability issues are not addressed. Cialdea Mayer et al. [16] provide a first formal characterization of flexible timelines and plans independent of temporal networks and Cialdea Mayer and Orlandini [15] address the controllability issue.

As far as controllability is concerned, previous works have tackled the issue of robust plan execution within a Constraint-based Temporal Planning (CBTP) framework, deploying specialized techniques based on temporal-constraint networks [23, 26, 27, 35]. Controllability issues have been formalized and investigated for Simple Temporal Problems with Uncertainty by Vidal and Fargier [35],

who give the basic formal notions properly defining *dynamic* controllability. Several authors (e.g., Hunsberger [23] and Morris [26]) have proposed a *dispatchable execution* approach, where a flexible temporal plan is used by a plan executive that schedules activities on-line while guaranteeing constraint satisfaction. These notions have also been extended to a timeline-based framework by Cesta et al. [11], exploiting a method based on model checking with Timed Game Automata (TGA) to check flexible plans against dynamic controllability and to generate a robust plan controller able to execute flexible timeline-based plans [29, 30]. Although following the same TGA-based approach, Cialdea Mayer and Orlandini [15] advance such previous work, providing a formalization for flexible timelines with uncertainty (not considered in the above mentioned approach) and including information about controllability in plan representation.

The contribution of this work is, up to our knowledge, the first comprehensive formalization of flexible timelines with uncertainty. It combines, extends and revises the characterization of flexible timelines and plans presented by Cialdea Mayer et al. [16] and Cialdea Mayer and Orlandini [15].The paper provides a general semantics for timeline-related planning concepts such as domains, goals, problems, constraints and flexible plans, taking into account the difference between controllable and uncontrollable activities. Uncertainty is taken into account in the characterization of valid plans, that are assumed not to take decisions on components the planner cannot control. The main controllability properties for flexible plans are finally defined.

Differently from all the previous works, which rely on external structures to represent timelines and, in particular, on temporal networks to characterize and address the controllability issue (apart from Cimatti et al. [17], with the limitations mentioned above), a key contribution of this work is to provide a formal stand-alone definition of the main concepts of timeline-based planning and the relative controllability properties, that is independent from the concrete structure exploited to represent timelines. In this way, controllability issues can be addressed independently from both the exploited plan representation and the solving engine. The importance of this feature is due to the fact that representing a flexible timeline-based plan as a temporal network entails a sort of simplification of the associated plan structure, thus causing a loss of information on the causal/temporal "dependencies" among its components. Such information can be useful for planning engines (as shown, for instance, by Umbrico et al. [33]) and, in general, for supporting a more detailed analysis of the relevant features enclosed in the generated plans. As a final remark, it is worth underscoring that, to the best of our knowledge, a comprehensive investigation of the formal relationship between temporal networks and timelines, either in terms of expressivity or complexity, is missing. This is an interesting, non trivial, open issue and the work presented here constitutes a first mandatory step to address such an issue in the future.

The paper is organized as follows. Section 2 provides the basic notions underlying timeline-based planning: state variables, flexible timelines and their schedules. The components of a planning domain are introduced in Section 3, presenting, in particular, the forms of constraints that may be imposed on the behavior of different system components. Section 4 defines flexible plans in such a way that they contain all the information needed for their execution. Planning problems and solutions plans are presented in Section 5, highlighting the fact that a well-defined planning problem must include information about the behavior of external

state variables. The standard notions of controllability are defined and discussed in Section 6. Finally, Section 7 concludes the paper providing a brief analysis of the adequacy of the proposed formalization to capture the frameworks underlying the approaches adopted by related P&S systems, a final discussion, and an overview of future work.

2 State Variables and Timelines

The timeline-based approach to planning pursues the general idea that planning and scheduling for controlling complex physical systems consist of the synthesis of desired temporal behaviors (or *timelines*). According to this paradigm, a domain is modeled as a set of features with an associated set of temporal functions on a finite set of values. The time-varying features are usually called *multi-valued state variables* [28]. Like in classical control theory, the evolution of the features is described by some causal laws and limited by domain constraints. These are specified in a *domain specification*. The task of a planner is to find a sequence of decisions that brings the timelines into a final desired set, satisfying the domain specification and special conditions called *goals*. Causal and temporal constraints specify which value transitions are allowed, the minimal and maximal duration of each valued interval and (so-called) *synchronization constraints* between different state variables.

In real-world domains, once a temporally flexible plan is generated, it has to be executed by an executive system that manages controllable processes in the presence of exogenous events. In this scenario, the execution process is not completely under the control of the executive. Thus, a mandatory requirement for dealing with P&S in real-world contexts is to distinguish between controllable and uncontrollable tasks. This work considers two sources of uncertainty. On the one hand, the evolution of some components of the system may be completely outside the control of the executive; such components are modeled by means of *external* state variables. What the planner and the executive know about them is only what is specified in the underlying planning problem. The distinction between external and *planned* state variables is part of the domain specification. On the other hand, some events may be only partially controllable: the system can decide when to start an activity, but is not allowed to fix its duration exactly. When the duration of a value cannot be controlled, it is tagged as *uncontrollable*, and what the planner and the executive may assume is only that its duration is included within given lower and upper bounds.

This section is devoted to properly define the basic notions modelling the system components and their temporal behavior (i.e., state variables, flexible timelines,[1] and non-flexible ones).

For the sake of generality, temporal instants and durations are taken from an infinite set of non negative numbers $\mathbb{T}$, including 0. For instance, $\mathbb{T}$ can be the set of natural numbers $\mathbb{N}$ (in a discrete time framework), as well as the non-negative real numbers $\mathbb{R}_{\geq 0}$. Sometimes, ∞ is given as an upper bound to allowed numeric values, with the meaning that $t < \infty$ for every $t \in \mathbb{T}$. The notation $\mathbb{T}^{\infty}$ will be

---

[1] In this work, "timeline" refers to what Cialdea Mayer et al. [16] call a flexible timeline, while non-flexible timelines are called "scheduled timelines". The latter are defined as particular cases of (flexible) timelines.

used to denote $\mathbb{T} \cup \{\infty\}$, $\mathbb{T}_{>0} = \mathbb{T} - \{0\}$ and $\mathbb{T}_{>0}^{\infty} = \mathbb{T}^{\infty} - \{0\}$. When dealing with temporal intervals, if $s, e \in \mathbb{T}$, the (closed) interval $[s, e]$ denotes the set of time points $\{t \mid s \leq t \leq e\}$.

In order to support the formal definitions given below, a simple case study will be used as a running example. The domain is inspired by a space mission long-term planning problem as described by Cesta et al. [12]. The mission consists of a remote satellite operating around a target planet. The satellite can either point to the planet and use its instruments to produce scientific data, e.g., take pictures with an infrared camera, or point towards a communication station, e.g., an Earth ground station, and communicate previously-produced data. It can also perform some maintenance operations. A set of operative constraints must be satisfied. For instance, the satellite sleweing between two different pointing states takes some time, and communication with Earth must occur within a ground-station availability window.

2.1 State Variables

A state variable x is characterized by four components: the set V of values the variable may assume, a function T mapping each value $v \in V$ to the set of values that are allowed to follow v, a function γ tagging each value with information about its controllability, and a function D which may set upper and lower bounds on the duration of each variable value.

Definition 1 A *state variable* x, where x is a unique identifier, called the *variable name*, is a tuple (V, T, γ, D), where:

1. V, also denoted by values(x), is a non-empty set, whose elements are the state variable *values*.
2. $T : V \rightarrow 2^V$ is a total function, called the state variable *value transition function*.
3. $\gamma : V \rightarrow \{c, u\}$ is a total function, called the *controllability tagging function*; $\gamma(v)$ is the *controllability tag* of the value v. If $\gamma(v) = c$, then v is a *controllable value*, and if $\gamma(v) = u$, then v is *uncontrollable*.
4. $D : V \rightarrow \mathbb{T} \times \mathbb{T}^{\infty}$ is a total function such that $D(v) = (d_{min}, d_{max})$ for some $d_{min} \geq 0$ and $d_{max} \geq d_{min}$, and if $\gamma(v) = u$, then $d_{min} > 0$ and $d_{max} \neq \infty$; D is called the state variable *duration function*.

If $\gamma(v) = c$, then the value v is under complete control of the system, which can exactly decide when to start and finish, i.e., to execute, the corresponding activity. When $\gamma(v) = u$, the value v models an activity whose exact duration cannot be controlled by the system (i.e., it is under the control of the environment).

The intuition behind the duration function is that if $D(v) = (d_{min}, d_{max})$, then the duration of each interval in which x has the value v is included between d_{min} and d_{max} inclusive, if $d_{max} \in \mathbb{T}$; it is not shorter than d_{min} and has no upper bound, if $d_{max} = \infty$.

In practice, existing systems, such as, for instance, EUROPA [3] and APSI-TRF [10], allow values to be represented by means of parametrized expressions and constraints on the parameters of adjacent values can be imposed. In the present theoretical approach, values are taken to be completely instantiated in order to

simplify the presentation. This amounts to describing sets and functions by enumeration and does not diminish the expressive power.

In what follows, we assume that, whenever a set of state variables SV is considered, for every distinct pair x and y in SV, $x \neq y$. Moreover, we assume that the set SV is partitioned into two disjoints sets, SV_P, containing the *planned* state variables, and SV_E, the set of the *external* ones. Every value v of an external state variable is uncontrollable, i.e., $\gamma(v) = u$. An external variable represents a component of the "external world" that is completely outside the system control: the planner cannot decide when to start or end its activities. What is known about an external variable is specified in the planning problem. On the contrary, a planned state variable represents a component of the system that is under the control of the executive. Nevertheless, controllable sub-systems may also have uncontrollable activities (i.e., activities whose starting times can be decided by the executive, but their durations, and consequently their ending times, are not controllable). In other terms, the planner can decide when to start an uncontrollable activity of a planned variable (i.e., when the variable assumes an uncontrollable value), even if it cannot be precisely predicted how long it will last. In general, when an activity (either controllable or not) is preceded by an uncontrollable one, the system cannot control its start time, since it depends on the duration of the uncontrollable previous one.

Example 1 In our running example, the timeline-based specification identifies two state variables, that will be called pm (for "pointing mode") and gv (for "ground station visibility"), whose values and transitions are illustrated in Figure 1. Therefore the set of considered state variables is $SV = \{pm, gv\}$.

The state variable pm models the satellite's pointing sub-system which can be oriented towards the Earth, stay oriented to the Earth while communicating with the ground station, perform maintanance operations, slew to point towards a predefined target, or gather scientific data. We assume that the only activity with uncertain duration is communication. The state variable is $pm = (V_{pm}, T_{pm}, \gamma_{pm}, D_{pm})$, where:

- $V_{pm} = \{Earth, Comm, Maintenance, Science, Slewing\}$;
- T_{pm} is the value transition function such that
    $T_{pm}(Earth) = \{Comm, Maintenance, Slewing\}$,
    $T_{pm}(Comm) = \{Earth, Maintenance\}$,
    $T_{pm}(Maintenance) = \{Earth\}$,
    $T_{pm}(Science) = \{Slewing\}$,
    $T_{pm}(Slewing) = \{Earth, Science\}$;
- γ_{pm} is the controllability tagging function such that
    $\gamma_{pm}(Comm) = u$, and
    $\gamma_{pm}(Earth) = \gamma_{pm}(Maintenance) = \gamma_{pm}(Science) = \gamma_{pm}(Slewing) = c$;
- D_{pm} is the duration function such that
    $D_{pm}(Earth) = [1, \infty]$,
    $D_{pm}(Comm) = [30, 50]$,
    $D_{pm}(Science) = [36, 58]$,
    $D_{pm}(Maintenance) = [90, 90]$,
    $D_{pm}(Slewing) = [30, 30]$.

The state variable gv models the ground station visibility windows during the satellite orbit and it is the tuple $gv = (V_{gv}, T_{gv}, \gamma_{gv}, D_{gv})$, where:
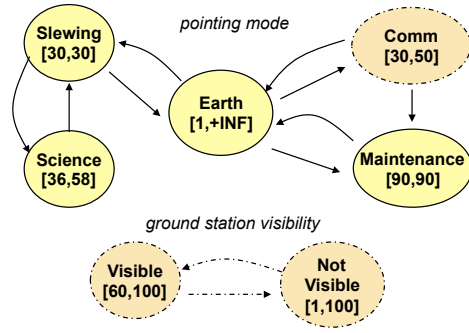
Fig. 1 State variables for the satellite domain. Orange (dotted) ovals represent uncontrollable values, yellow ones are controllable.

- $V_{gv} = \{Visible, NotVisible\}$;
- the function γ_{gv} tags both values $Visible$ and $NotVisible$ with u;
- T_{gv} is such that
  $T_{gv}(Visible) = \{NotVisible\}$
  $T_{gv}(NotVisible) = \{Visible\}$;
- D_{gv} is the duration function such that
  $D_{gv}(Visible) = [60, 100]$
  $D_{gv}(NotVisible) = [1, 100]$.

In the planning domain representing this system, pm is a planned state variable, while gv is an external one: it is completely uncontrollable and the system is not allowed to take decisions on its behavior. Therefore $SV_P = \{pm\}$ and $SV_E = \{gv\}$.

2.2 Timelines

A timeline represents the temporal evolution of a system component up to a given time. It is made up of a sequence of valued intervals, called *tokens*, each of which represents a time slot in which the variable assumes a given value. For instance, Figure 2-a, on the left, represents a timeline made up of three tokens: the first one, lasting from 0 to 25, has the value Earth; the following token, with value Slewing, begins at time point 25 and ends at 55; the last one, with value Science, starts at 55 and ends at 100. When planning with timelines, time flexibility is taken into account by allowing token durations to range within given bounds. For instance, Figure 2-b, on the right, represents a flexible timeline, where the exact times at which tokens begin and end are replaced by time intervals; the first token may end at a time ranging between 23 and 28, etc. The notion of (flexible) timeline is introduced in the following definition.

Definition 2 If $x = (V, T, \gamma, D)$ is a state variable, a *token* for the variable x has the form:

$$x^i = (v, [e, e'], [d, d'], \gamma(v))$$

where x^i, for $i \in \mathbb{N}$, is the token *name*, $v \in V$, $e, e', d, d' \in \mathbb{T}$, $e \leq e'$ and $d_{min} \leq d \leq d' \leq d_{max}$, for $D(v) = (d_{min}, d_{max})$. The value $\gamma(v)$ is called the token
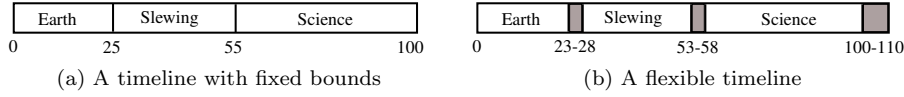
(a) A timeline with fixed bounds   (b) A flexible timeline

Fig. 2 Representation of two timelines

controllability tag; if $\gamma(v) = c$, then the token is *controllable*; if $\gamma(v) = u$, then it is an *uncontrollable* token.

A *timeline FTL_x* for the state varible $x = (V, T, \gamma, D)$ is a finite sequence of tokens for x, of the form:

$$x^1 = (v_1, [e_1, e_1'], [d_1, d_1'], \gamma(v_1)), \ldots, x^k = (v_k, [e_k, e_k'], [d_k, d_k'], \gamma(v_k)),$$

where for all $i = 1 \ldots k-1$, $v_{i+1} \in T(v_i)$ and $e_i' \leq e_{i+1}$. The interval $[e_k, e_k']$ in the last token is called the *horizon* of the timeline and the number k of tokens making up FTL_x is its *length*.

If $x^i = (v, [e, e'], [d, d'], \gamma(v))$ is a token in the timeline FTL_x, then:

- $\mathrm{val}(x^i) = v$;
- $\mathrm{end\_time}(x^i) = [e, e']$;
- $\mathrm{start\_time}(x^0) = [0, 0]$ and $\mathrm{start\_time}(x^{i+1}) = \mathrm{end\_time}(x^i)$;
- $\mathrm{duration}(x^i) = [d, d']$;
- with an abuse of notation, $\gamma(x^i)$ denotes the token controllability tag $\gamma(\mathrm{val}(x^i))$.

Intuitively, a token x^i of the above form represents the set of valued intervals starting at some $s \in \mathrm{start\_time}(x^i)$, ending at some $e \in \mathrm{end\_time}(x^i)$ and whose durations are in the range $\mathrm{duration}(x^i)$. The horizon of the timeline is the end time of its last token.

Example 2 Let us consider the timeline FTL_{pm} for the state variable pm, in the satellite case study, made of the following sequence of tokens:

$$pm^1 = (Earth, [23, 28], [20, 30], c)$$
$$pm^2 = (Slewing, [53, 58], [30, 30], c)$$
$$pm^3 = (Science, [100, 110], [45, 58], c)$$

Figure 2-b gives a graphical representation of FTL_{pm} (where duration bounds and controllability tags are not repesented). The horizon of FTL_{pm} is $[100, 110]$.

A timeline for pm is represented in Figure 2-a. It is made of the following sequence of tokens:

$$pm^1 = (Earth, [25, 25], [20, 30], c)$$
$$pm^2 = (Slewing, [55, 55], [30, 30], c)$$
$$pm^3 = (Science, [100, 100], [45, 58], c)$$

and its horizon is $[100, 100]$.

It is worth pointing out that often in the literature (e.g., [16, 21]) a flexible token contains also a *start* interval. However, once a token x^i is embedded in a timeline, the time interval to which its start point belongs ($\mathrm{start\_time}(x^i)$) can

easily be computed as shown in the definition above. Thus, including it as part of the token itself is redundant.

On the contrary, duration restrictions alone would be inadequate to precisely identify when the valued intervals represented by a given token must begin and end. As a matter of fact, duration and end time bounds interact when determining which legal values a token end time may assume. Let us assume, for instance, that the duration of a given token x^i is $[20, 30]$ and that one may compute, from the durations of the previous tokens, that its start time is $[40, 50]$. One can then infer that the end points of the valued intervals it represents are necessarily in the range $[60, 80] = [40 + 20, 50 + 30]$. However, it may be the case that a stricter end time is required, for instance $[65, 75]$. In this case, starting x^i at 50 and ending it at 80, though respecting the duration bounds, would not be a legal value to "execute" the token, since $80 \notin [65, 75]$. So, differently from the case of non-flexible timelines, durations alone are not sufficient to suitably represent tokens. Analogously, end time bounds do not capture all the necessary information: the above described token x^i does not represent a valued interval starting at 40 and ending at 75, since, though it respects the start and end time bounds, it violates the duration constraint.

Controllability tags are among the components of token structures for a different reason. Although $\gamma(x^i)$ is equal to $\gamma(\text{val}(x^i))$, such information is included in the token x^i with the aim of having a self-contained representation of flexible plans, encapsulating all the relevant execution information. This allows the executive system to handle plans with no need of considering also the description of the state variables.

When considering a set **FTL** of timelines for the state variables in SV, it is always assumed that it contains exactly one timeline for each element of SV.

2.3 Schedules

A *scheduled timeline* is a particular case where each token has a singleton $[t, t]$ as its end time, i.e., the end times are all fixed. A *schedule* of a timeline FTL_x is essentially obtained from FTL_x by narrowing down to singletons (time points) the tokens end times. This subsection is devoted to define such notions. The schedule of a token corresponds to one of the valued intervals it represents (i.e., it is obtained by choosing an exact end point in the allowed interval, without changing its duration bounds). A scheduled timeline is a sequence of scheduled tokens satisfying the duration requirements. Tokens, timelines and sets of timelines represent the set of their schedules. In general, STL_x and **STL** will be used as meta-variables for scheduled timelines and sets of scheduled timelines, respectively, while FTL_x and **FTL** as meta-variables for generic (flexible) timelines and sets of timelines.

In what follows, an interval of the form $[t, t]$, consisting of a single time point, will be identified with the time point t (and, with an abuse of notation, singleton intervals are allowed as operands of additions, subtractions, comparison operators, etc.).

Definition 3 A *scheduled* token is a token of the form $x^i = (v, [t, t], [d, d'], \gamma(v))$ (or succinctly $x^i = (v, t, [d, d'], \gamma(v))$). A *schedule* of a token $x^i = (v, [e, e'], [d, d'], \gamma(v))$ is a scheduled token $x^i = (v, t, [d, d'], \gamma(v))$, where $e \leq t \leq e'$.

A *scheduled timeline* STL_x is a timeline consisting only of scheduled tokens and such that if k is the timeline length, then: for all $1 \leq i \leq k$, if $\text{duration}(x^i) = [d_i, d'_i]$, then $d_i \leq \text{end\_time}(x^i) - \text{start\_time}(x^i) \leq d'_i$.

A scheduled timeline STL_x for the state variable x is a *schedule* of FTL_x if STL_x and FTL_x have the same length k, and for all i, $1 \leq i \leq k$, the token x^i of STL_x is a schedule of the token x^i of FTL_x.

Let **FTL** be a set of timelines for the state variables in SV. A schedule **STL** of **FTL** is a set of scheduled timelines for the state variables in SV, where each $STL_x \in$ **STL** is a schedule of the timeline $FTL_x \in$ **FTL**.

In simple terms, a scheduled timeline is a timeline where every end time is a singleton respecting the duration bounds. A schedule of a timeline is a way of assigning values to each token end time, so that both duration and end time bounds are respected. Tokens, timelines, and sets of timelines represent the set of their respective schedules.

Example 3 Let us consider the flexible timeline FTL_{pm} of example 2:

$$FTL_{pm} = pm^1 = (Earth, [23, 28], [20, 30], c)$$
$$pm^2 = (Slewing, [53, 58], [30, 30], c)$$
$$pm^3 = (Science, [100, 110], [45, 58], c)$$

It is worth pointing out that, since the start time of the first token of a timeline is $[0, 0]$, its end time bounds are usually equal to its duration bounds, but, like this example shows, it is not necessarily so. When the two intervals differ, the end point of the corresponding first token in any schedule of the timeline belongs to their intersection.

Each schedule of the timeline FTL_{pm} represents a series of choices for the token end points, within the allowed intervals and respecting the allowed durations. For instance, the following timeline is a schedule of FTL_{pm}:

$$STL_{pm} = pm^1 = (Earth, 25, [20, 30], c)$$
$$pm^2 = (Slewing, 55, [30, 30], c)$$
$$pm^3 = (Science, 105, [45, 58], c)$$

In fact, it satisfies all of the endpoint and duration bounds in FTL_{pm}.

Clearly, not every sequence of scheduled tokens is a scheduled timeline. For instance, the sequence of tokens obtained from STL_{pm} by replacing the token pm^2 with $pm^2 = (Slewing, 58, [30, 30], c)$ is not a scheduled timeline at all, since it does not satisfy the duration constraints for pm^2: $\text{end\_time}(pm^2) - \text{start\_time}(pm^2) = 58 - 25 = 33 > 30$.

Let us now consider the scheduled timeline STL'_{pm} obtained from STL_{pm} by replacing the token pm^3 with $pm^3 = (Science, 113, [45, 58], c)$. Although STL'_{pm} satisfies all the duration bounds, it is not a schedule of FTL_{pm}, since the end time of pm^3 is not in the allowed interval $[100, 110]$ of pm^3 in FTL_{pm}.

3 Temporal Relations, Synchronization Rules, and Planning Domains

The behavior of state variables may be restricted by requiring that time intervals with given state variable values satisfy some temporal constraints. For instance,

in our sample domain, the satellite can communicate its data to the Earth only during a ground station visibility window. In other terms, for every token pm^i in the timeline for the state variable pm having the value $Comm$, there must exist a token in the timeline for gv, with the value $Visible$ and bearing a given temporal relation with pm^i. This section is devoted to see how such *synchronization rules* can be expressed, thus completing the definition of all of the components of a domain specification.

3.1 Temporal Relations

As a first step, the set of allowed temporal relations is introduced. They are either relations between two intervals or relations between an interval and a time point. In this work, quantitative temporal constraints are considered and, for the sake of simplicity, a small set of primitive relations is chosen, all of which are parametrized by a (single) temporal interval. Other constraints corresponding to the quantitative extension of Allen's temporal relations between intervals [2] and those used by systems like EUROPA [3] and APSI-TRF [10], can be easily defined in terms of the primitive ones, as it will be shown below.

Definition 4 A *temporal relation between intervals* is an expression of the form $A \rho_{[lb,ub]} B$, where $A = [s_A, e_A]$ and $B = [s_B, e_B]$ are time intervals, with s_A, $e_A, s_B, e_B \in \mathbb{T}$, $\rho \in \mathbf{R} = \{$ start_before_start, end_before_end, start_before_end, end_before_start$\}$, $lb \in \mathbb{T}$ and $ub \in \mathbb{T}^\infty$. The following table defines when a relation $A \rho_{[lb,ub]} B$ holds:

the relation	holds if
A start_before_start$_{[lb,ub]}$ B	$lb \leq s_B - s_A \leq ub$
A end_before_end$_{[lb,ub]}$ B	$lb \leq e_B - e_A \leq ub$
A start_before_end$_{[lb,ub]}$ B	$lb \leq e_B - s_A \leq ub$
A end_before_start$_{[lb,ub]}$ B	$lb \leq s_B - e_A \leq ub$

A *temporal relation between an interval and a timepoint* is an expression of the form $A \rho_{[lb,ub]} t$, where $A = [s, e]$ is a time interval, with $s, e \in \mathbb{T}$, $\rho \in \mathbf{R}' = \{$starts_before, starts_after, ends_before, ends_after$\}$, $t, lb \in \mathbb{T}$, and $ub \in \mathbb{T}^\infty$. The following table defines when a relation $A \rho_{[lb,ub]} t$ holds:

the relation	holds if
A starts_before$_{[lb,ub]}$ t	$lb \leq t - s \leq ub$
A starts_after$_{[lb,ub]}$ t	$lb \leq s - t \leq ub$
A ends_before$_{[lb,ub]}$ t	$lb \leq t - e \leq ub$
A ends_after$_{[lb,ub]}$ t	$lb \leq e - t \leq ub$

Other relations can be defined in terms of the primitive ones, like those in Table 1 (and their converses). In the following, when the relations in the leftmost column of the table are used, they are meant as abbreviations of the corresponding expressions on their right.

Once relations on time intervals are defined, they can be transposed to relations on tokens. The expressions used to denote such relations refer to tokens by means of their names.

the relation	is defined as
A equals B	A start_before_start$_{[0,0]}$ B and
	A end_before_end$_{[0,0]}$ B
A meets B	A end_before_start$_{[0,0]}$ B
A before$_{[lb,ub]}$ B	A end_before_start$_{[lb,ub]}$ B
A overlaps$_{[lb_1,ub_1][lb_2,ub_2]}$ B	A start_before_start$_{[lb_1,ub_1]}$ B and
	A end_before_end$_{[lb_2,ub_2]}$ B and
	B start_before_end$_{[0,\infty]}$ A
A contains$_{[lb_1,ub_1][lb_2,ub_2]}$ B	A start_before_start$_{[lb_1,ub_1]}$ B and
	B end_before_end$_{[lb_2,ub_2]}$ A
A starts$_{[lb,ub]}$ B	A start_before_start$_{[0,0]}$ B and
	A end_before_end$_{[lb,ub]}$ B
A finishes$_{[lb,ub]}$ B	A start_before_start$_{[lb,ub]}$ B and
	A end_before_end$_{[0,0]}$ B
A starts_at t	A starts_before$_{[0,0]}$ t
A ends_at t	A ends_before$_{[0,0]}$ t

Table 1 Defined temporal relations

Definition 5 Let x^i and y^j be names of tokens belonging to scheduled timelines for the state variables x and y, respectively, with start_time$(x^i) = s_i$, end_time$(x^i) = e_i$, start_time$(y^j) = s_j$, end_time$(y^j) = e_j$. Moreover, let $t, lb \in \mathbb{T}$ and $ub \in \mathbb{T}^\infty$. Expressions of the form $x^i \, \rho_{[lb,ub]} y^j$, for $\rho \in \mathbf{R}$, and $x^i \, \rho_{[lb,ub]} t$, for $\rho \in \mathbf{R}'$, are called *relations on tokens*. The relation $x^i \, \rho_{[lb,ub]} y^j$ holds iff $[s_i, e_i] \, \rho_{[lb,ub]} [s_j, e_j]$ holds and the relation $x^i \, \rho_{[lb,ub]} t$ holds iff $[s_i, e_i] \, \rho_{[lb,ub]} t$ holds. When a relation on two tokens x^i and y^j holds, we also say that the tokens x^i and y^j satisfy the relation, and that any set of scheduled timelines that contain x^i and y^j satisfies the relation. Analogously, if a relation $x^i \, \rho_{[lb,ub]} t$ holds, then the token x^i and any set of scheduled timelines containing x^i satisfy the relation.

Example 4 Let $\mathbf{STL} = \{STL_{pm}, STL_{gv}\}$ be a set of timelines for the satellite domain, where STL_{pm} contains the tokens

$$pm^5 = (Earth, 100, [1, 43], c)$$
$$pm^6 = (Comm, 140, [30, 50], u)$$

and STL_{gv} contains the tokens

$$gv^1 = (NotVisible, 60, [60, 80], u)$$
$$gv^2 = (Visible, 130, [50, 90], u)$$

The expressions gv^2 start_before_start$_{[5,\infty]}$ pm^6 and pm^6 ends_before$_{[20,30]}$ 165 are relations on tokens and they are satisfied by $\mathbf{STL}$.

3.2 Synchronizations

A synchronization constraint can be informally thought as a statement of the form "for every token … there exist tokens such that …". In other words, it represents a kind of quantified sentence. The formal counterpart of this kind of assertions makes use of variables: for every var_0 … there exist $var_1, … var_n$ such that ….

The variables used to express synchronizations are called *token variables*. They are taken from a (potentially infinite) set $X = \{a_0, a_1, \dots\}$ of names, whose elements are all different from variable names, values and numbers. These variables are intended to range over tokens in the considered set of timelines. Making a step forward, it can be observed that synchronization assertions actually use a form of bounded quantification: "for all/exist tokens *with value v* in the timeline for the state variable $x \dots$". Such token variables with restricted range will be denoted by expressions of the form $a_i[x = v]$, where a_i is a token variable, x is a state variable name, and $v \in \text{values}(x)$. Such expression are called *annotated token variables*.

The next definition introduces the form of the assertions that can be used to express parametrized relations on tokens.

Definition 6 An *atom* is either the special constant $\top$ or an expression of the form $a_i\, \rho_{[lb,ub]}\, a_j$ or $a_i\, \rho'_{[lb,ub]}\, t$, where a_i and a_j are token variables, $lb, t \in \mathbb{T}$, $ub \in \mathbb{T}^\infty$, $\rho \in \mathbf{R}$, and $\rho' \in \mathbf{R}'$.

An *existential statement* is an expression of the form

$$\exists\, a_1[x_1 = v_1] \dots a_n[x_n = v_n]\, . \, \mathcal{C}$$

where

(i) $a_1, \dots, a_n$ are distinct token variables;
(ii) for all $i = 1, \dots, n$, x_i is a state variable and $v_i \in \text{values}(x_i)$ (i.e., $a_i[x_i = v_i]$ is an annotated token variable);
(iii) $\mathcal{C}$ is a conjunction of atoms.

The *bound variables* of the statement are $a_1, \dots, a_n$ and any variable different from $a_1, \dots, a_n$ possibly occurring in $\mathcal{C}$ is said to occur *free* in the statement.

Disjunctions of existential statements constitute the body of synchronization rules.

Definition 7 A *synchronization rule* is an expression of the form

$$a_0[x_0 = v_0] \rightarrow \mathcal{E}_1 \vee \cdots \vee \mathcal{E}_k$$

(for $k \geq 1$) where every $\mathcal{E}_i$ is an existential statement whose bound variables are all different from a_0 and where only the token variable a_0 may occur free. The left-hand part of the synchronization rule, $a_0[x_0 = v_0]$, is called the *trigger* of the rule.

A synchronization rule with empty trigger is an expression of the form:

$$\top \rightarrow \mathcal{E}_1 \vee \cdots \vee \mathcal{E}_k$$

(for $k \geq 1$) where every $\mathcal{E}_i$ is an existential statement with no free variables.

Intuitively, a synchronization rule with non-empty trigger of the above form requires that, whenever the state variable x_0 assumes the value v_0 in some interval a_0, there is at least an existential statement $\mathcal{E}_i = \exists\, a_1[x_1 = v_1] \dots a_n[x_n = v_n]\,.\,\mathcal{C}$ and tokens a_i ($1 \leq i \leq n$) where the variable x_i has the value v_i, such that $\mathcal{C}$ holds. (If $\mathcal{C} = \top$, no temporal relation is required to hold.) When the trigger is empty, the existence of the intervals a_i and the relations among them have to hold unconditionally. Synchronization rules with empty triggers are useful to represent

domain invariants, as well as planning goals (both called "facts" by Cimatti et al. [17]). The use of token variables (which are absent in Cimatti et al. [17]) allows one to refer to different intervals having the same value. Indeed, although the token variables $a_0, \ldots, a_n$ are pairwise distinct, multiple occurrences of state variable names and values are allowed.

It is worth pointing out that a more compact formulation of synchronization rules can be obtained by allowing $\mathcal{C}$ to be an arbitrary combination of atoms by means of both conjunctions and disjunctions (as in Cialdea Mayer et al. [16]). However, this would require a planner to be able to transform positive boolean formulae into disjunctive normal form, in order to obtain the set of possible choices (conjunctions of atoms) implying that $\mathcal{C}$ holds (i.e., the implicants of $\mathcal{C}$).

Example 5 Consider the sample operational constraint in the satellite domain seen at the beginning of this section: the satellite communicates with the Earth only when the ground station is visible. A synchronization rule expressing this constraint is the following:

$$a_0[pm = Comm] \to \exists\, a_1[gv = Visible] \,.\, a_1 \text{ contains}_{[0,\infty][0,\infty]} a_0$$

According to this rule, whenever the state variable *pm* assumes the value *Comm* in an interval a_0, the state variable *gv* has the value *Visible* in some interval a_1 containing a_0.

Synchronization rules with empty triggers may be useful to state known facts, such as, for instance:

$$\top \to \exists a_1[pm = Earth].\ a_1 \text{ starts\_at } 0$$

This rule represents the fact that, at the beginning, the satellite is locked on the Earth.

Synchronization rules with empty triggers are also used to represent planning goals, as will be described later on.

The following definition introduces the semantics of synchronization on scheduled timelines. Since the statement of a synchronization rule makes use of token variables, each of them must be "interpreted", i.e., mapped to a token of the considered timelines.

Definition 8 Let **FTL** be a set of timelines for the state variables SV and $\{x_1, \ldots, x_n\} \subseteq SV$. A *token assignment* for a set of annotated token variables $\{a_1[x_1 = v_1], \ldots, a_n[x_n = v_n]\}$ on **FTL** is a function φ mapping every a_i to a token of the timeline $FTL_{x_i} \in \textbf{FTL}$ and such that $\text{val}(\varphi(a_i)) = v_i$ for all $i = 1, \ldots, n$.

Let $\mathcal{C} = A_1 \wedge \cdots \wedge A_m$ be a conjunction of atoms and **STL** a set of scheduled timelines, including a timeline for every state variable occurring in $\mathcal{C}$. A token assignment φ on **STL** satisfies $\mathcal{C}$ if for every atom $A \in \{A_1, \ldots, A_m\}$,

(i) if $A = a_i\, \rho_{[lb,ub]} a_j$ then the relation $\varphi(a_i)\, \rho_{[lb,ub]} \varphi(a_j)$ holds;
(ii) if $A = a_i\, \rho_{[lb,ub]} t$, then the relation $\varphi(a_i)\, \rho_{[lb,ub]} t$ holds.

A token assignment φ on **STL** satisfies an existential statement of the form $\exists\, a_1[x_1 = v_1] \ldots a_n[x_n = v_n]\,.\,\mathcal{C}$ if φ is a token assignment for a set of annotated variables including $a_0[x_0 = v_0], \ldots, a_n[x_n = v_n]$ and φ satisfies $\mathcal{C}$.

Let $S = a_0[x_0 = v_0] \to \mathcal{E}_1 \vee \cdots \vee \mathcal{E}_k$ be a synchronization rule. A set of scheduled timelines **STL** for the state variables SV *satisfies the synchronization rule* S if for every token x_0^k in $STL_{x_0} \in$ **STL** such that $\mathrm{val}(x_0^k) = v_0$, there exists a token assignment φ on **STL** such that $\varphi(a_0) = x_0^k$ and φ satisfies $\mathcal{E}_i$ for some $i \in \{1, \ldots, k\}$.

A set of timelines **STL** for the state variables SV satisfies a synchronization rule with empty trigger $\top \to \mathcal{E}_1 \vee \cdots \vee \mathcal{E}_k$ if, for some $i \in \{1, \ldots, k\}$, there exists a token assignment φ on **STL** satisfying $\mathcal{E}_i$.

Let SV be a set of state variables, $\mathcal{S}$ be a set of synchronization rules concerning variables in SV and **STL** be a set of scheduled timelines for the state variables in SV. **STL** *satisfies the set of synchronizations* $\mathcal{S}$ iff **STL** satisfies all the elements of $\mathcal{S}$.

Example 6 Consider the synchronization rule given in Example 5, that constrains the satellite to communicate only when the ground station is visible:

$$a_0[pm = Comm] \to \exists\, a_1[gv = Visible]\,.\ a_1 \ \mathrm{contains}_{[0,\infty][0,\infty]} a_0$$

Let STL_{pm} and STL_{gv} be two scheduled timelines. Assume that the timeline for the pointing system contains a single token whose value is $Comm$, and has the form:

$$STL_{pm} = \ldots, pm^{i-1} = (Earth, 90, [50, 80], c),\ pm^i = (Comm, 130, [30, 50], u),\ \ldots$$

In addition, assume that the timeline STL_{gv} for the ground station visibility contains the token $gv^j = (Visible, 170, [60, 100], u)$, with $\mathrm{start\_time}(gv^j) = 80$.

The synchronization rule is satisfied by the set of scheduled timelines $\{STL_{pm}, STL_{gv}\}$. Indeed, pm^i is the only token in STL_{pm} whose value is $Comm$, and the token assignment φ, such that $\varphi(a_0) = pm^i$ and $\varphi(a_1) = gv^j$, satisfies the existential statement $\exists\, a_1[gv = Visible]\,.\ a_1 \ \mathrm{contains}_{[0,\infty][0,\infty]} a_0$: $\mathrm{val}(\varphi(a_1)) = Visible$ and φ satisfies $a_1 \ \mathrm{contains}_{[0,\infty][0,\infty]} a_0$. The latter assertion holds because $\varphi(a_1) \ \mathrm{contains}_{[0,\infty][0,\infty]} \varphi(a_0)$ – i.e., $gv^j \ \mathrm{contains}_{[0,\infty][0,\infty]} pm^i$ – holds, since $[80, 170] \ \mathrm{contains}_{[0,\infty][0,\infty]} [90, 130]$ holds.

3.3 Planning Domains

A planning domain is described by specifying a set of state variables and a set of synchronization rules. The formal definition of planning domains is given next, together with the notion of a set of scheduled timelines respecting the requirements of the domain.

Definition 9 A *planning domain* is a triple $(SV_P, SV_E, \mathcal{S})$, where:

- SV_P is a set of planned state variables;
- SV_E is a set of external state variables (with $SV_P \cap SV_E = \varnothing$);
- $\mathcal{S}$ is a set of synchronization rules involving state variables in $SV_P \cup SV_E$.

A set of scheduled timelines **STL** for the state variables in SV is *valid* with respect to the planning domain $\mathcal{D} = (SV_P, SV_E, \mathcal{S})$ if $SV = SV_P \cup SV_E$ and **STL** satisfies the set of synchronizations $\mathcal{S}$.

Example 7 Let us consider the planning domain $\mathcal{D} = (SV_P, SV_E, \mathcal{S}\}$ where $SV_P = \{pm\}$, $SV_E = \{gv\}$ – for the state variables pm and gv of example 1 – and $\mathcal{S}$ contains only the synchronization rule of examples 5 and 6: $a_0[pm = Comm] \rightarrow \exists\, a_1[gv = Visible]\,.\, a_1$ contains$_{[0,\infty][0,\infty]}\, a_0$.

Let moreover **STL** be the set of timelines containing

$$
\begin{aligned}
STL_{pm} = pm^1 &= (Earth, 90, [50, 80], c)\\
pm^2 &= (Comm, 130, [30, 50], u)\\
pm^3 &= (Earth, 200, [80, 130], c)\\
STL_{gv} = gv^1 &= (NotVisible, 80, [70, 100], u)\\
gv^2 &= (Visible, 170, [60, 100], u)\\
gv^3 &= (NotVisible, 200, [10, 50], u)
\end{aligned}
$$

The set **STL** is valid with respect to the planning domain $\mathcal{D}$: it contains exactly one timeline for each state variable in $SV_P \cup SV_E$ and it satisfies the synchronization rule of the domain, as shown in example 6.

4 Flexible Plans

The main component of a flexible plan is a set **FTL** of timelines, representing different sets **STL**$_i$ of scheduled ones. It may be the case that not every **STL**$_i$ satisfies the synchronization rules of the domain. We aim at defining plans so that they contain all the information needed to execute them, without having to check how the behavior of state variables and timelines is constrained by the planning domain.[2] Consequently, in order to guarantee that every set of scheduled timelines represented by a given flexible plan Π (i.e., the different ways of executing Π) is valid with respect to the underlying planning domain, the plan has to be equipped with additional information about the temporal relations that have to hold in order to satisfy the synchronization rules of the domain.

As a schematic example showing why a set of timelines does not convey enough information to represent a flexible plan, let us consider a domain with a synchronization rule S of the form $a_0[x = v] \rightarrow \exists a_1[y = v']\,.\, a_0$ meets a_1 and timelines for the state variables x and y containing, respectively, the tokens $x^i = (v, [30, 50], [20, 30], \gamma(v))$ and y^j, with val$(y^j) = v'$ and start_time$(y^j) = [30, 50]$. Not every pair of schedules of x^i and y^j satisfies S. Thus, the representation of a flexible plan must also include information about the relations that must hold between tokens in order to satisfy the synchronization rules of the planning domain. In the example above, it would include the relation x^i meets y^j. In general, a flexible plan includes a set of relations on tokens. When there are different ways to satisfy a synchronization rule by the same set **FTL** of flexible timelines, there are also different (valid) flexible plans with the same set of timelines **FTL**: each of them represents a different way to satisfy the synchronizations. A flexible plan represents the set of its "instances": the schedules of its timelines which satisfy the relations on tokens required by the plan.

Definition 10 A *flexible plan* Π is a pair (**FTL**, $\mathcal{R}$), where **FTL** is a set of timelines and $\mathcal{R}$ is a set of relations on tokens, involving token names in some timelines in **FTL**.

---

[2] For the same reason controllability tags are included in token descriptions.

An *instance* of the flexible plan $\Pi = (\mathbf{FTL}, \mathcal{R})$ is any schedule of $\mathbf{FTL}$ that satisfies every relation in $\mathcal{R}$.

Example 8 Let $\Pi = (\mathbf{FTL}, \mathcal{R})$, where:

– $\mathbf{FTL}$ contains the timelines

$$
\begin{aligned}
FTL_{pm} = \; &pm^1 = (Earth, [50, 70], [50, 70], c) \\
&pm^2 = (Comm, [80, 120], [30, 50], u) \\
&pm^3 = (Earth, [200, 200], [80, 120], c) \\
FTL_{gv} = \; &gv^1 = (NotVisible, [70, 90], [70, 90], u) \\
&gv^2 = (Visible, [130, 190], [60, 100], u) \\
&gv^3 = (NotVisible, [200, 210], [10, 70], u)
\end{aligned}
$$

– $\mathcal{R} = \{pm^2 \; \text{end\_before\_end}_{[30,50]} \; gv^2\}$

Π is a flexible plan, and the set $\mathbf{STL}$ of scheduled timelines containing

$$
\begin{aligned}
STL_{pm} = \; &pm^1 = (Earth, 70, [50, 70], c) \\
&pm^2 = (Comm, 120, [30, 50], u) \\
&pm^3 = (Earth, 200, [80, 120], c) \\
STL_{gv} = \; &gv^1 = (NotVisible, 80, [70, 90], u) \\
&gv^2 = (Visible, 170, [60, 100], u) \\
&gv^3 = (NotVisible, 205, [10, 70], u)
\end{aligned}
$$

is an instance of Π, since $\{STL_{pm}, STL_{gv}\}$ is a schedule of $\mathbf{FTL}$ and it satisfies the relation $pm^2 \; \text{end\_before\_end}_{[30,50]} \; gv^2$, since pm^2 ends 50 time points before the end of gv^2.

If however the end time of pm^2 in STL_{pm} is replaced by 100, the so obtained set of scheduled timelines is not an instance of Π, although it is a schedule of $\mathbf{FTL}$, because the relation in $\mathcal{R}$ is not satisfied.

In order to determine when a plan is valid with respect to a planning domain, the semantics of synchronizations on flexible plans must be defined. Essentially, a plan $\Pi = (\mathbf{FTL}, \mathcal{R})$ satisfies a synchronization rule S if the constraints represented by S are guaranteed to hold for any schedule of $\mathbf{FTL}$ satisfying the relations in $\mathcal{R}$. In other terms, $\mathcal{R}$ represents a possible way to satisfy S. The intuition underlying the formal definition can be explained as follows. When considering a plan $\Pi = (\mathbf{FTL}, \mathcal{R})$, a mapping is used to assign the annotated token variables occurring in the synchronization to token names occurring in $\mathbf{FTL}$. Let us consider, for instance a rule of the form $a_0[x = v] \to \exists a_1[y = v'].a_1 \; \text{end\_before\_start}_{[10,20]} \; a_0$. Let us moreover assume that x^3 is a token in the timeline for x in $\mathbf{FTL}$ with $\text{val}(x^3) = v$, and that the timeline for y in $\mathbf{FTL}$ contains exactly two tokens y^5 and y^8 having value v'. In order for the rule to be satisfied, $\mathbf{FTL}$ must be constrained by requiring that x^3 starts from 10 to 20 time units after the end of either y^5 or y^8. The plan Π commits to one of the two alternatives: binding a_1 to either y^5 or y^8. The correspondence between token variables and token names is established by use of a function φ mapping a_1 to either y^5 or y^8. According to the chosen option, the set of relations $\mathcal{R}$ in the plan contains either $y^5 \; \text{end\_before\_start}_{[10,20]} \; x^3$ or $y^8 \; \text{end\_before\_start}_{[10,20]} \; x^3$.

Definition 11 Let $\mathcal{C} = A_1 \wedge \cdots \wedge A_m$ be a conjunction of atoms, $\Pi = (\mathbf{FTL}, \mathcal{R})$ a flexible plan, where $\mathbf{FTL}$ contains a timeline for every state variable occurring in $\mathcal{C}$, and φ a token assignment on $\mathbf{FTL}$. The plan Π satisfies $\mathcal{C}$ with φ if for every atom $A \in \{A_1, \ldots, A_m\}$, (i) if $A = a_i \, \rho_{[lb,ub]} a_j$, then $\varphi(a_i) \, \rho_{[lb,ub]} \varphi(a_j) \in \mathcal{R}$, and (ii) if $A = a_i \, \rho_{[lb,ub]} t$, then $\varphi(a_i) \, \rho_{[lb,ub]} t \in \mathcal{R}$.

Let $\mathcal{E} = \exists a_1[x_1 = v_1] \ldots a_n[x_n = v_n] . \mathcal{C}$ be an existential statement and φ be a token assignment on $\mathbf{FTL}$. The flexible plan Π satisfies $\mathcal{E}$ with φ if φ is an assignment for a set of annotated token variables including $a_0[x_0 = v_0], \ldots, a_n[x_n = v_n]$ and Π satisfies $\mathcal{C}$ with φ.

The plan Π satisfies a synchronization rule with non-empty trigger

$$a_0[x_0 = v_0] \to \mathcal{E}_1 \vee \cdots \vee \mathcal{E}_k$$

if for every flexible token x_0^m of the timeline $FTL_{x_0} \in \mathbf{FTL}$ such that $\mathrm{val}(x_0^m) = v_0$, there exists a token assignment φ on $\mathbf{FTL}$ such that $\varphi(a_0) = x_0^m$ and Π satisfies $\mathcal{E}_i$ with φ, for some $i \in \{1, \ldots, k\}$.

The plan Π satisfies a synchronization rule with empty trigger

$$\top \to \mathcal{E}_1 \vee \cdots \vee \mathcal{E}_k$$

if, for some $i \in \{1, \ldots, k\}$, there exists a token assignment φ on $\mathbf{FTL}$ such that Π satisfies $\mathcal{E}_i$ with φ.

Example 9 Let us consider a flexible timeline FTL_{pm} for the state variable pm representing the evolution of the pointing sub-system through the states $Slewing - Science - Slewing - Earth - Comm - Earth$, whose fifth token (the only one where the satellite is in communication mode) is:

$$pm^5 = (Comm, [120, 130], [30, 50], u)$$

with $\mathrm{start\_time}(pm^5) = [70, 80]$. Moreover, let FTL_{gv} be a timeline containing the token

$$gv^4 = (Visible, [110, 170], [60, 100], u)$$

with $\mathrm{start\_time}(gv^4) = [10, 70]$. The flexible plan $\Pi = (\mathbf{FTL}, \mathcal{R})$, where $\mathbf{FTL} = \{FTL_{pm}, FTL_{gv}\}$ and

$$\mathcal{R} = \{gv^4 \, \mathrm{start\_before\_start}_{[0,\infty]} \, pm^5, pm^5 \, \mathrm{end\_before\_end}_{[0,\infty]} \, gv^4\}$$

satisfies the synchronization rule of Example 5. Considering in fact the definition of the relation contains given in Table 1, the rule can be rewritten as:

$$a_0[pm = Comm] \to \exists a_1[gv = Visible] . (a_1 \, \mathrm{start\_before\_start}_{[0,\infty]} \, a_0)$$
$$\wedge (a_0 \, \mathrm{end\_before\_end}_{[0,\infty]} \, a_1)$$

The set $\mathcal{R}$ contains both atoms

$$\varphi(a_1) \, \mathrm{start\_before\_start}_{[0,\infty]} \, \varphi(a_0)$$
$$\text{and } \varphi(a_0) \, \mathrm{end\_before\_end}_{[0,\infty]} \, \varphi(a_1),$$

for the token assignment φ such that $\varphi(a_0) = pm^5$ and $\varphi(a_1) = gv^4$.

Clearly, there might be schedules of the set of timelines $\mathbf{FTL}$ that do not satisfy the requirement $gv^4 \, \mathrm{contains}_{[0,\infty][0,\infty]} \, pm^5$. For example, the requirement

is not satisfied by schedules where $\text{start\_time}(pm^5) = 80$, $\text{end\_time}(pm^5) = 130$, $\text{start\_time}(gv^4) = 50$ and $\text{end\_time}(gv^4) = 120$, which consequently are not instances of the flexible plan Π. However, if, for example, $\text{start\_time}(pm^5) = 80$, $\text{end\_time}(pm^5) = 130$, $\text{start\_time}(gv^4) = 50$ and $\text{end\_time}(gv^4) = 150$, the requirement gv^4 contains$_{[0,\infty][0,\infty]}$ pm^5 is satisfied, thus schedules of **FTL** with such tokens are also instances of Π.

As a further example showing how a plan commits to a choice among the possibly different ways to satisfy a synchronization rule, let us consider a set **FTL** of timelines and a rule S of the form

$$a_0[x = v] \rightarrow \exists a_1[y = v'].\, a_1 \text{ end\_before\_start}_{[10,20]} a_0$$
$$\vee \exists a_1[z = v''].\, a_0 \text{ end\_before\_start}_{[5,\infty]} a_1$$

Let us moreover assume that x^3 and x^7 are the only tokens with value v in the timeline FTL_x for x in **FTL**, that the timeline FTL_y for y in **FTL** contains exactly one token y^5 having value v', and that FTL_z contains exactly one token z^8 with value v''.

In order to satisfy the rule S:

1. $\mathcal{R}$ must contain either y^5 end_before_start$_{[10,20]}$ x^3 or x^3 end_before_start$_{[5,\infty]}$ z^8. In fact, the plan has to satisfy either (i) $\exists a_1[y = v'].\, a_1$ end_before_start$_{[10,20]}$ a_0 or (ii) $\exists a_1[z = v''].\, a_0$ end_before_start$_{[5,\infty]}$ a_1 with a token assignment φ such that $\varphi(a_0) = x^3$. If $\mathcal{R}$ contains y^5 end_before_start$_{[10,20]}$ x^3, then the plan satisfies the existential statement (i) with φ, when $\varphi(a_1) = y^5$. If it contains x^3 end_before_start$_{[5,\infty]}$ z^8, then the plan satisfies (ii) with φ, when $\varphi(a_1) = z^8$.
2. $\mathcal{R}$ must contain either y^5 end_before_start$_{[10,20]}$ x^7 or x^7 end_before_start$_{[5,\infty]}$ z^8: the reasoning is the same as above, just replacing x^7 for x^3.

Therefore, for instance, the plans

$$(\mathbf{FTL}, \{y^5 \text{ end\_before\_start}_{[10,20]} x^3, x^7 \text{ end\_before\_start}_{[5,\infty]} z^8\})$$
$$\text{and } (\mathbf{FTL}, \{x^3 \text{ end\_before\_start}_{[5,\infty]} y^5, y^5 \text{ end\_before\_start}_{[10,20]} x^7\})$$

satisfy S.

The notions of plan validity and consistency can finally be defined.

Definition 12 A flexible plan $\Pi = (\mathbf{FTL}, \mathcal{R})$ is *valid* with respect to a planning domain $\mathcal{D} = (SV_P, SV_E, \mathcal{S})$ iff:

1. **FTL** is a set of timelines for the state variables $SV = SV_P \cup SV_E$;
2. Π satisfies all the synchronization rules in $\mathcal{S}$;
3. for each planned state variable $x = (V, T, \gamma, D) \in SV_P$, and each uncontrollable token x^i in $FTL_x \in \mathbf{FTL}$, if $D(\text{val}(x^i)) = (d_{min}, d_{max})$ and $\text{start\_time}(x^i) = [s, s']$, then $\text{duration}(x^i) = [d_{min}, d_{max}]$ and $\text{end\_time}(x^i) = [s + d_{min}, s' + d_{max}]$.

The plan Π is *consistent* if there exists at least one instance of Π.

The last condition required for a plan to be valid guarantees that the plan does not make any hypothesis on the duration of uncontrollable values of planned variables. The restriction is not applied to external variables, since the planner

is not allowed to control them at all: their behavior is described in the planning problem as a sort of observation of the external world.

Plan consistency is a minimal requirement for a plan to be considered meaningful, although, when the domain includes uncontrollable elements, it is not enough to guarantee its executability. Cialdea Mayer et al. [16] prove a result showing that there exists a set Θ of flexible plans for which an effective consistency check procedure exists, and yet every scheduled valid plan is an instance of some flexible plan in Θ. Intuitively, each plan $\Pi \in \Theta$ is such that the sequence of scheduled tokens obtained by fixing every token end point to the lower bound of the respective end time interval is an instance of Π (i.e., it is a scheduled timeline respecting the relations in Π). The mentioned result implies that, when searching for a consistent plan, it is sufficient to consider candidate plans in Θ, respecting the above condition.

5 Planning Problems and Solution Plans

5.1 Planning Goals

A planning problem includes the description of the underlying planning domain and of a desired goal to be accomplished. This work considers *temporally extended goals*: a planning goal specifies that some planned variables have to assume some given values in some intervals and, possibly, that such intervals must satisfy some temporal relations. Disjunctive goals are also allowed.

Definition 13 A *planning goal* $\mathcal{G}$ for a domain $\mathcal{D} = (SV_P, SV_E, \mathcal{S})$ is a pair (Γ, Δ), where:

(i) Γ is a set of *accomplishment goals*, i.e., expressions of the form $g = (x, v)$, where g is a token variable, called the goal name, $x \in SV_P$, and $v \in \text{values}(x)$;

(ii) Δ, called a *relational goal*, is a disjunction $\mathcal{D}_1 \vee \cdots \vee \mathcal{D}_k$, where each $\mathcal{D}_i$ is a conjunction of atoms containing only goal names occurring in Γ.

A planning goal $\mathcal{G} = (\Gamma, \Delta)$, with $\Gamma = \{g_1 = (x_1, v_1), \ldots, g_n = (x_n, v_n)\}$ and $\Delta = \mathcal{D}_1 \vee \cdots \vee \mathcal{D}_k$, is represented by a synchronization rule $S_{\mathcal{G}}$ with empty trigger, of the form:

$$\top \to \exists g_1[x_1 = v_1] \ldots g_n[x_n = v_n]. \mathcal{D}_1 \vee \cdots \vee \exists g_1[x_1 = v_1] \ldots g_n[x_n = v_n]. \mathcal{D}_k$$

It is worth pointing out that restrictions on the start and end intervals of a given goal (like in [10, 17]) can be expressed by means of relational goals. In particular, if the start point of a given goal g is required to be in the interval $[s, s']$ and its end point in $[e, e']$, then such restrictions can be expressed by the relational goal $(g \text{ starts\_after}_{[0, s'-s]} s) \wedge (g \text{ ends\_after}_{[0, e'-e]} e)$.

Example 10 A simple planning goal for the satellite domain may be that, in order to accomplish the mission, the satellite has to perform a scientific operation and then communicate the results to the ground station no later than 65 time units after the completion of the scientific task. It is represented by the pair (Γ, Δ), where $\Gamma = \{g_1 = (pm, Science), g_2 = (pm, Comm)\}$ and $\Delta = g_1 \text{ end\_before\_start}_{[0,65]} g_2$; which can be turned into the synchronization rule

$$\top \to \exists g_1[pm = Science] \, g_2[pm = Comm] \, . \, g_1 \text{ end\_before\_start}_{[0,65]} g_2$$

Analogously, if the satellite has to perform a communication and a maintenance operation and we want to specify an alternative ordering constraint between them, i.e., the maintenance operation is accomplished either no more than 90 time units before or immediately after the communication task, the goal is $\mathcal{G} = (\Gamma, \Delta)$, where $\Gamma = \{g_1 = (pm, Comm), g_2 = (pm, Maintenance)\}$, and $\Delta = (g_2 \text{ end\_before\_start}_{[0,90]} g_1) \vee (g_1 \text{ meets } g_2)$. The corresponding synchronization rule is:

$$\top \rightarrow \exists\, g_1[pm = Comm]\ g_2[pm = Maintenance]\,.\,g_2 \text{ end\_before\_start}_{[0,90]} g_1$$
$$\vee\ \exists\, g_1[pm = Comm]\ g_2[pm = Maintenance]\,.\,g_1 \text{ meets } g_2$$

The next definition introduces the notion of goal fulfillment for scheduled timelines.

Definition 14 A set of scheduled timelines **STL** fulfills the planning goal $\mathcal{G}$ if it satisfies the synchronization rule $S_\mathcal{G}$ representing $\mathcal{G}$.

5.2 Planning Problems

In timeline-based planning, a planning problem typically includes a *planning horizon*, i.e., the time by which the system behavior has to be planned. Finally, since the external state variables are not under the system control, the problem must include information about their behavior up to the given horizon. Such information is given in the form of a set of flexible timelines and temporal relations on their tokens.

Definition 15 A *planning problem* is a tuple $(\mathcal{D}, \mathcal{G}, \mathcal{O}, H)$, where $\mathcal{D} = (SV_P, SV_E, \mathcal{S})$ is a planning domain, $\mathcal{G}$ a planning goal for $\mathcal{D}$, $H \in \mathbb{T}_{>0}$ is the planning horizon, and $\mathcal{O} = (\mathbf{FTL}_E, \mathcal{R}_E)$, where

(i) $\mathbf{FTL}_E$ is a set containing exactly one flexible timeline for each external state variable in SV_E;
(ii) the horizon of every timeline in $\mathbf{FTL}_E$ is $[h, h']$ for some $h \geq H$;
(iii) $\mathcal{R}_E$ is a set of temporal relations on tokens of timelines in $\mathbf{FTL}_E$;
(iv) $(\mathbf{FTL}_E, \mathcal{R}_E)$ is consistent, i.e., there is at least one schedule of $\mathbf{FTL}_E$ satisfying the relations in $\mathcal{R}_E$.

The pair $\mathcal{O}$, called the *observation*, specifies the behavior of external state variables up to a time point not less than the planning horizon. Item (iv) rules out inconsistent observations, i.e., descriptions of the behavior of the external state variables with no instances. The pair $(\mathbf{FTL}_E, \mathcal{R}_E)$ can be viewed as a flexible plan. In particular, even when $\mathcal{R}_E = \varnothing$, the set of timelines $\mathbf{FTL}_E$ must have at least one schedule. The planner must respect what is specified by the set of timelines $\mathbf{FTL}_E$, without taking any autonomous decision: this requirement is fulfilled simply when the timeline for each external variable in the plan is exactly the timeline for the same state variable in $\mathbf{FTL}_E$. The relations in $\mathcal{R}_E$ represent known facts about the external world.[3] During the planning phase, they are actually ignored, but they are taken into account by the executor, which is allowed to assume that the

---

[3] Cialdea Mayer and Orlandini [15] do not allow constraints on the behavior of external state variables.

behavior of the external state variables is restricted so that the constraints in $\mathcal{R}_E$ are respected (see Section 6).

It is worth pointing out that, since $\mathbf{FTL}_E$ is a set of timelines, the planner knows how the external components evolve, i.e., the sequence of activities/states constituting their behavior, the only uncertainty being the duration of such states. This rules out, for instance, scenarios where the uncontrollable events might occur an unknown number of times within the given horizon.

Example 11 For instance, a planning problem for our sample domain can be the problem $\Pi = (\mathcal{D}, \mathcal{G}, \mathcal{O}, H)$, where

- $\mathcal{D}$ is the planning domain of example 7, i.e., $\mathcal{D} = (SV_P, SV_E, \mathcal{S}\}$ where $SV_P = \{pm\}$, $SV_E = \{gv\}$ – for the state variables pm and gv of example 1 – and $\mathcal{S}$ contains only the synchronization rule

$$a_0[pm = Comm] \to \exists\, a_1[gv = Visible]\,.\ a_1 \text{ contains}_{[0,\infty][0,\infty]} a_0;$$

- $\mathcal{G} = (\Gamma, \Delta)$, where
  $\Gamma = \{g_1 = (pm, Science),\ g_2 = (pm, Comm)\}$ and
  $\Delta = g_1 \text{ before}_{[0,65]} g_2$ (as in example 10);
- $\mathcal{O} = (\{FTL_{gv}\}, \emptyset)$, where $FTL_{gv} =$

$$gv^1 = (NotVisible, [60, 80], [60, 80], u)$$
$$gv^2 = (Visible, [150, 160], [80, 100], u)$$
$$gv^3 = (NotVisible, [250, 260], [90, 100], u)$$

- $H = 250$.

5.3 Solution Plans

Definition 16 Let $\mathcal{P} = (\mathcal{D}, \mathcal{G}, \mathcal{O}, H)$ be a planning problem and $\Pi = (\mathbf{FTL}, \mathcal{R})$ be a flexible plan. Π is a *flexible solution plan* for $\mathcal{P}$ if:

1. for every planned state variable x, the horizon of $FTL_x \in \mathbf{FTL}$ is $[H, H]$;
2. Π is valid with respect to $\mathcal{D}$;
3. Π satisfies the synchronization rule $S_{\mathcal{G}}$ representing $\mathcal{G}$;
4. If $\mathcal{O} = (\mathbf{FTL}_E, \mathcal{R}_E)$, then $\mathbf{FTL}_E \subseteq \mathbf{FTL}$.

The first condition above guarantees that the behavior of the planned state variables is determined exactly up to the horizon of the planning problem, henceforth (condition 3) all the planning goals are achieved in due time. It is worth pointing out that condition 1 implies that the last token of each planned timeline must be controllable. Condition 4 ensures that the plan does not make any assumption on external variables, except for what is implied by the state variable definition and the observation.

Example 12 Let us consider, for instance, the problem $\mathcal{P}$ of example 11 and the flexible plan $\Pi = (\mathbf{FTL}, \mathcal{R})$, where:

- **FTL** $= \{FTL_{gv}, FTL_{pm}\}$, where FTL_{gv} is the timeline of example 11 and FTL_{pm} is the following timeline:

$$pm^1 = (Earth, [1, 33], [1, 33], c)$$
$$pm^2 = (Slewing, [31, 63], [30, 30], c)$$
$$pm^3 = (Science, [67, 99], [36, 58], c)$$
$$pm^4 = (Slewing, [97, 129], [30, 30], c)$$
$$pm^5 = (Earth, [98, 140], [1, 43], c)$$
$$pm^6 = (Comm, [128, 190], [30, 50], u)$$
$$pm^7 = (Earth, [250, 250], [90, 122], c)$$

- $\mathcal{R}$ contains the two relations on tokens pm^3 end_before_start$_{[0,65]}$ pm^6 and $gv^2 contains_{[0,\infty],[0,\infty]} pm^6$.

The plan Π is a flexible solution plan for the planning problem $\mathcal{P}$ because:

- the horizon of FTL_{pm} is 250;
- Π is valid with respect to $\mathcal{D}$:
  - **FTL** contains the two timelines for pm and gv;
  - Π satisfies the synchronization rule of the domain: the only token in FTL_{pm} with value $Comm$ is pm^6 and $\mathcal{R}$ contains $gv^2 contains_{[0,\infty],[0,\infty]} pm^6$, where the value of gv^2 is $Visible$;
  - the only uncontrollable token in FTL_{pm} is pm^6, $D_{pm}(Comm) = [30, 50] =$ duration(pm^6), start_time$(pm^6) = [98, 140]$, and end_time$(pm^6) = [128, 190]$ $= [98 + 30, 140 + 50]$.
- Π satisfies the synchronization rule $S_{\mathcal{G}}$ representing $\mathcal{G}$: the tokens pm^3 and pm^6 have values $Science$ and $Comm$, respectively, and $\mathcal{R}$ contains the relation pm^3 before$_{[0,65]}$ pm^6.
- $FTL_{gv} \in$ **FTL**.

The next result proves that information encoded by a flexible solution plan Π for a given planning problem is sufficient to ensure that every instance of Π is valid with respect to the planning domain and it fulfills the goal. Although the proof of this result is a straightforward consequence of the definitions, it deserves to be stated explicitly, since flexible plans without such a property would be meaningless.

Theorem 1 *If the plan Π is a flexible solution plan for the problem $\mathcal{P} = (\mathcal{D}, \mathcal{G}, \mathcal{O}, H)$, then every instance of Π is valid with respect to $\mathcal{D}$ and fulfills the goal $\mathcal{G}$.*

Proof Since the state variables of a flexible plan Π and any of its instances are the same, it is sufficient to show that if Π satisfies a synchronization rule S (according to Definition 11), then every instance of Π satisfies S (according to Definition 8). This implies also that any instance of Π fulfills the planning goal $\mathcal{G}$, because Π satisfies the synchronization rule $S_{\mathcal{G}}$ representing $\mathcal{G}$.

Let us assume that $\Pi = (\mathbf{FTL}, \mathcal{R})$ satisfies the synchronization rule

$$S = a_0[x_0 = v_0] \rightarrow \mathcal{E}_1 \vee \cdots \vee \mathcal{E}_k$$

and **STL** is an instance of Π.

Let x_0^m be a token of the timeline $STL_{x_0} \in$ **STL**, with val$(x_0^m) = v_0$. The corresponding token x_0^m in $FTL_{x_0} \in$ **FTL** has the same value v_0. Since Π satisfies

S, there exists a token assigment φ, with $\varphi(a_0) = x_0^m$, such that Π satisfies $\mathcal{E}_i$ with φ for some $i \in \{1, \ldots, k\}$. Consequently, if $\mathcal{E}_i = \exists\, a_1[x_1 = v_1] \ldots a_n[x_n = v_n]\, .\mathcal{C}$, φ is a token assignment for a set of annotated token variables including $a_1[x_1 = v_1], \ldots, a_n[x_n = v_n]$ (i.e., $\mathrm{val}(\varphi(a_i)) = v_i$ for all $i = 1, \ldots, n$) and Π satisfies $\mathcal{C}$ with φ. This means that, if $\mathcal{C} = A_1 \wedge \cdots \wedge A_m$, for every atom $A \in \{A_1, \ldots, A_m\}$:

(i) if $A = a_i\, \rho[lb, ub]\, a_j$, then $\varphi(a_i)\, \rho[lb, ub]\, \varphi(a_j) \in \mathcal{R}$; since **STL** is an instance of Π, it satisfies the relation $\varphi(a_i)\, \rho[lb, ub]\, \varphi(a_j)$.
(ii) If $A = a_i\, \rho[lb, ub]\, t$, then $\varphi(a_i)\, \rho[lb, ub]\, t \in \mathcal{R}$, and, as above, **STL** satisfies $\varphi(a_i)\, \rho[lb, ub]\, t$.

As a consequence, the token assignment φ on **STL** satisfies $\mathcal{C}$ and the same assignment satisfies the existential statement $\mathcal{E}_i$. Since $\varphi(a_0) = x_0^m$, **STL** satisfies the synchronization rule S. The case of synchronization rules with empty triggers is treated similarly. $\qquad\square$

6 Controllability Properties of Flexible Plans

Once a flexible plan $\Pi = (\mathbf{FTL}, \mathcal{R})$ is built, controllability tags and the set of temporal constraints on tokens of external state variables given in the planning problem are the important features to be taken into consideration when facing the controllability problem for flexible plans. Controllability issues have been addressed representing a flexible plan as a Simple Temporal Network with Uncertainty (STNU) [35] . In order to transpose the same concepts into our framework, it can be observed that token end points correspond to nodes in the network, while token durations and the temporal constraints given in the sets of relations $\mathcal{R}$ contained in the plan and in the observations $\mathcal{R}_E$ given by the planning problem correspond to network edges. Durations of uncontrollable tokens and relations in $\mathcal{R}_E$ correspond to what are usually called *contingent links* in a STNU.

It is worth pointing out that, although disjunctions are allowed in the formulation of synchronization rules, disjunctive constraints do not occur in a flexible plan. i.e., the set of relations of a plan represents a commitment to exactly one of the possible ways to satisfy the synchronizations.

This section proposes a definition of the various notions of plan controllability, very much in the style of similar works on STNU [22, 27, 35]. Some flaws present in the corresponding definitions given by Cialdea Mayer and Orlandini [15] are also corrected.

In what follows, if **FTL** is a set of timelines, $\mathrm{tokens}(\mathbf{FTL})$ denotes the set of all the tokens making up the timelines in **FTL**, $\mathrm{tokens}_C(\mathbf{FTL})$ is the set of controllable tokens occurring in some timeline in **FTL**, and $\mathrm{tokens}_U(\mathbf{FTL})$ contains the uncontrollable tokens of **FTL**.

The notion of *situation*, introduced next, copes with the temporal uncertainty represented by the uncontrollable tokens of a set of timelines **FTL**. A situation is a function assigning a (legal) value to the duration of each uncontrollable token. The set of situations defined over a set **FTL** of timelines represents all the associated uncontrollable temporal evolutions. Every situation ω for **FTL** induces a set of timelines where the duration of every uncontrollable token x^i in **FTL** is replaced by the (singleton) value $\omega(x^i)$. In other terms, if x^i is an uncontrollable token of the

form $(v, [e, e'], [d, d'])$, then it is replaced in $\omega(\mathbf{FTL})$ by $(v, [e, e'], \omega(x^i))$; controllable tokens are left unchanged. The set of timelines obtained in this way is called a *projection* of $\mathbf{FTL}$: in a projection, the duration of each uncontrollable token is fixed. Intuitively, a projection corresponds to one of the possible combinations of uncontrollable behaviors in $\mathbf{FTL}$.

Definition 17 Let $\mathbf{FTL}$ be a set of timelines and ω a total function

$$\omega : \text{tokens}_U(\mathbf{FTL}) \to \mathbb{T}$$

If $FTL_x \in \mathbf{FTL}$, the *projection* $\omega(FTL_x)$ is the timeline obtained from FTL_x by replacing the duration of every uncontrollable token x^i with $[\omega(x^i), \omega(x^i)]$. The projection $\omega(\mathbf{FTL})$ is the set $\{\omega(FTL_x) \mid FTL_x \in \mathbf{FTL}\}$.

A function $\omega : \text{tokens}_U(\mathbf{FTL}) \to \mathbb{T}$ is a *situation* for $\mathbf{FTL}$ iff:

1. for every $x^i \in \text{tokens}_U(\mathbf{FTL})$, with $\text{duration}(x^i) = [d, d']$, $d \le \omega(x^i) \le d'$;
2. every timeline in the projection $\omega(\mathbf{FTL})$ has at least one schedule.

The set of all the situations for $\mathbf{FTL}$ is called the *space of situations* for $\mathbf{FTL}$ and is denoted by $\Omega_{\mathbf{FTL}}$.

It is worth observing that if $(\mathbf{FTL}, \mathcal{R})$ is a flexible plan, then $(\omega(\mathbf{FTL}), \mathcal{R})$ is a flexible plan too.

Example 13 Let us consider the set $\mathbf{FTL}$ of timelines containing:

$$\begin{aligned}
FTL_x = x^1 &= (v_1, [100, 150], [100, 150], c), \\
x^2 &= (v_2, [130, 180], [20, 30], u), \\
x^3 &= (v_3, [200, 200], [20, 70], c) \\
FTL_y = y^1 &= (v'_1, [130, 180], [130, 180], u), \\
y^2 &= (v'_2, [200, 240], [20, 70], u)
\end{aligned}$$

Let moreover ω be the function such that $\omega(x^2) = 25$, $\omega(y^1) = 150$ and $\omega(y^2) = 20$. Its projection $\omega(\mathbf{FTL})$ is the set of timelines containing:

$$\begin{aligned}
\omega(FTL_x) = x^1 &= (v_1, [100, 150], [100, 150], c), \\
x^2 &= (v_2, [130, 180], 25, u), \\
x^3 &= (v_3, [200, 200], [20, 70], c) \\
\omega(FTL_y) = y^1 &= (v'_1, [130, 180], 150, u), \\
y^2 &= (v'_2, [200, 240], 20, u)
\end{aligned}$$

The function ω satisfies condition 1 of Definition 17. However, since the only way to fix the end points of tokens in $\omega(FTL_y)$ in order to satisfy the duration constraints is to set $\text{end\_time}(y^1) = 150$ and $\text{end\_time}(y^2) = 170 \notin [200, 240]$, the timeline $\omega(FTL_y)$ has no schedules. As a consequence, ω is not a situation for $\mathbf{FTL}$.

On the contrary, the function ω' with $\omega'(x^2) = 25$, $\omega'(y^1) = 150$ and $\omega'(y^2) = 50$ is a situation for $\mathbf{FTL}$. In fact, its projection

$$\begin{aligned}
\omega'(FTL_x) = x^1 &= (v_1, [100, 150], [100, 150], c), \\
x^2 &= (v_2, [130, 180], 25, u), \\
x^3 &= (v_3, [200, 200], [20, 70], c) \\
\omega'(FTL_y) = y^1 &= (v'_1, [130, 180], 150, u), \\
y^2 &= (v'_2, [200, 240], 50, u)
\end{aligned}$$

has at least one schedule.

As shown in example 13, if x is an external state variable and FTL_x is a timeline for x, then $\omega(FTL_x)$ has only one schedule: since all the tokens in FTL_x are uncontrollable, once their durations are exactly fixed, their end times are also fixed. In what follows, the projection $\omega(FTL_x)$ of an external state variable will be identified with its single schedule. As a consequence, it can be determined whether a given set of projections of timelines for external state variables satisfies a relation on tokens. In general, it can be determined whether a projection $\omega(\mathbf{FTL})$ satisfies a relation on tokens of external state variables.

When the planning problem contains information on some temporal relations that are known to hold between the tokens of external state variables (the component $\mathcal{R}_E$ of the observation), the executor is allowed to assume that the actual evolution of external components will respect such constraints. As a consequence, it can restrict the space of situations for $\mathbf{FTL}$ to those satisfying such constraints, that are called *relevant* to the problem.

Definition 18 If $\mathcal{P} = (\mathcal{D}, \mathcal{G}, \mathcal{O}, H)$ ia a planning problem, with $\mathcal{O} = (\mathbf{FTL}_E, \mathcal{R}_E)$, a situation ω for $\mathbf{FTL}$ is *relevant* to $\mathcal{P}$ if $\omega(\mathbf{FTL})$ satisfies all the relations in $\mathcal{R}_E$.

The subset of $\Omega_{\mathbf{FTL}}$ containing the situations relevant to $\mathcal{P}$ (the space of relevant situations) will be denoted by $\Omega_{\mathbf{FTL}, \mathcal{P}}$.

It is worth pointing out that, since the observation in a planning problem $\mathcal{P}$ is consistent (according to Definition 15), the set of situations relevant to $\mathcal{P}$ is not empty. In the following, the planning problem will sometimes be left implicit and we shall simply refer to "relevant situations".

There is a one-to-one correspondence between situations for a given set of timelines and their projections. Analogously, the set of schedules of a given set $\mathbf{FTL}$ of timelines bears a one-to-one correspondence with the set of functions assigning a (legal) fixed value to each token end time. Such functions are called *scheduling functions* and are defined next.

Definition 19 Let $\mathbf{FTL}$ be a set of timelines and θ a total function

$$\theta : \mathrm{tokens}(\mathbf{FTL}) \to \mathbb{T}$$

Such a function induces the set $\mathbf{STL}_\theta$ of sequences of scheduled tokens obtained from $\mathbf{FTL}$ by replacing the end time of each token $x^i \in \mathrm{tokens}(\mathbf{FTL})$ with $[\theta(x^i), \theta(x^i)]$.

A *scheduling function* for $\mathbf{FTL}$ is a function $\theta : \mathrm{tokens}(\mathbf{FTL}) \to \mathbb{T}$, such that $\mathbf{STL}_\theta$ is a set of scheduled timelines (i.e., each sequence of tokens in $\mathbf{STL}_\theta$ respects all the duration bounds). The set of all the scheduling functions for $\mathbf{FTL}$ is denoted by $\mathcal{T}_{\mathbf{FTL}}$.

Let $\Pi = (\mathbf{FTL}, \mathcal{R})$ be a flexible plan. A scheduling function θ for $\mathbf{FTL}$ is *consistent* with Π iff the set $\mathbf{STL}_\theta$ of scheduled timelines induced by θ is an instance of Π.

It is worth noticing that, while a situation fixes token durations (and maybe only indirectly their end times), a scheduling function assigns values to token end times. Moreover, situations are defined only on uncontrollable tokens, while scheduling functions are defined for all the tokens in the set of timelines.

Intuitively, a scheduling function that is consistent with the plan $\Pi = (\mathbf{FTL}, \mathcal{R})$ induces a set of scheduled timelines that respect all the duration requirements in $\mathbf{FTL}$ and all the relations in $\mathcal{R}$.

Example 14 Let us consider the set of timelines $\mathbf{FTL} = \{FTL_x, FTL_y\}$ of example 13:

$$FTL_x = x^1 = (v_1, [100, 150], [100, 150], c),$$
$$x^2 = (v_2, [130, 180], [30, 30], c),$$
$$x^3 = (v_3, [200, 200], [20, 70], c)$$
$$FTL_y = y^1 = (v'_1, [130, 180], [130, 180], u),$$
$$y^2 = (v'_2, [200, 240], [20, 70], u)$$

Let θ be the function such that $\theta(x^1) = 120$, $\theta(x^2) = \theta(y^1) = 150$, $\theta(x^3) = 200$, and $\theta(y^2) = 220$. θ is a scheduling function for $\mathbf{FTL}$, inducing the set $\mathbf{STL}_\theta$ of scheduled timelines containing:

$$STL_x = x^1 = (v_1, 120, [100, 150], c),$$
$$x^2 = (v_2, 150, [30, 30], c),$$
$$x^3 = (v_3, 200, [20, 70], c)$$
$$STL_y = y^1 = (v'_1, 150, [130, 180], u),$$
$$y^2 = (v'_2, 220, [20, 70], u)$$

Let now $\Pi = (\mathbf{FTL}, \mathcal{R})$ be a flexible plan, with $\mathcal{R} = \{x^2 \text{ meets } y^2\}$. The scheduling function θ is consistent with Π, since $\mathbf{STL}_\theta$ is an istance of Π: $\mathbf{STL}_\theta$ is a schedule of $\mathbf{FTL}$ (the duration constraints are satisfied by $\mathbf{STL}_\theta$, i.e., it is a set of scheduled timelines, and every token end time is in the allowed interval), and $\mathbf{STL}_\theta$ satisfies the relation $x^2 \text{ meets } y^2$.

Execution strategies are defined next. An execution strategy for a given plan Π maps every relevant situation to a scheduling function: once the duration of the uncontrollable tokens is known, the strategy decides how to schedule all the token end points.

Definition 20 Let $\Pi = (\mathbf{FTL}, \mathcal{R})$ be a flexible solution plan for the planning problem $\mathcal{P}$. An *execution strategy* for Π is a mapping $\sigma : \Omega_{\mathbf{FTL}, \mathcal{P}} \to \mathcal{T}_{\mathbf{FTL}}$.

The execution strategy σ is *viable* if for each relevant situation $\omega \in \Omega_{\mathbf{FTL}, \mathcal{P}}$, the scheduling function $\sigma(\omega)$ is consistent with Π and for every uncontrollable token x^i in the set of scheduled timelines induced by $\sigma(\omega)$, $\text{duration}(x^i) = \omega(x^i)$.

A viable strategy for the plan $(\mathbf{FTL}, \mathcal{R})$ maps each relevant situation ω to a scheduling function inducing a set of scheduled timelines $\mathbf{STL}$ such that:

- $\mathbf{STL}$ respects the end time and duration bounds in $\mathbf{FTL}$;
- $\mathbf{STL}$ satisfies the relations in $\mathcal{R}$;
- the duration of each uncontrollable token x^i in $\mathbf{STL}$ is equal to the exact duration of x^i given by ω.

Example 15 Let $\Pi = (\mathbf{FTL}, \mathcal{R})$ be the flexible plan of example 14 and let σ be the execution strategy mapping every (relevant) situation ω to the scheduling function θ such that $\theta(x^1) = \omega(y^1) - 30$, $\theta(x^2) = \omega(y^1)$, $\theta(x^3) = 200$, $\theta(y^1) = \omega(y^1)$, and $\theta(y^2) = \omega(y^1) + \omega(y^2)$. If ω is a situation, $\sigma(\omega)$ induces the set $\mathbf{STL}_\omega$ of scheduled timelines containing:

$$STL_x = x^1 = (v_1, \omega(y^1) - 30, [100, 150], c),$$
$$x^2 = (v_2, \omega(y^1), [30, 30], c),$$
$$x^3 = (v_3, 200, [20, 70], c)$$
$$STL_y = y^1 = (v'_1, \omega(y^1), [130, 180], u),$$
$$y^2 = (v'_2, \omega(y^1) + \omega(y^2), [20, 70], u)$$

In order to show that σ is viable it must be shown that for every situation ω:

1. the duration of every uncontrollable token x^i in $\mathbf{STL}_\omega$ is equal to $\omega(x^i)$. This is a consequence of the fact that $\text{duration}(y^1) = \text{end\_time}(y^1) = \omega(y^1)$ and $\text{duration}(y^2) = \text{end\_time}(y^2) - \text{end\_time}(y^1) = \omega(y^1) + \omega(y^2) - \omega(y^1) = \omega(y^2)$.
2. $\sigma(\omega)$ is consistent with $(\mathbf{FTL}, \mathcal{R})$, i.e., $\mathbf{STL}_\omega$ is an instance of $(\mathbf{FTL}, \mathcal{R})$. This holds because:
   - $\mathbf{STL}_\omega$ is a schedule of $\mathbf{FTL}$. First of all, we prove that $\mathbf{STL}_\omega$ is a set of scheduled timelines, i.e. each token end point complies with the respective duration constraints. Since, by hypothesis, ω is a situation, $\text{duration}(y^1) = \omega(y^1) \in [130, 180]$. Consequently, $\text{duration}(x^1) = \omega(y^1) - 30 \in [100, 150]$, $\text{duration}(x^3) = 200 - \omega(y^1) \in [20, 70]$. Moreover $\text{duration}(x^2) = \omega(y^1) - (\omega(y^1) - 30) \in [30, 30]$ and $\text{duration}(y^2) = \omega(y^2) \in [20, 70]$ (because situations respect duration bounds).
   The fact that each token end point in $\mathbf{STL}_\omega$ belongs to the respective end time interval in $\mathbf{FTL}$ follows similarly from the fact that, by hypothesis, $\omega(y^1) \in [130, 180]$.
   - $\mathbf{STL}_\omega$ satisfies the relation $x^2 \text{ meets } y^2$, because $\text{end\_time}(x^2) = \omega(y^1) = \text{start\_time}(y^1)$.

As a consequence, σ is a viable execution strategy.

In order to define dynamic execution strategies, i.e., strategies which are able to schedule a given event only on the base of what happened before, partial situations must be considered.

Definition 21 Let $\mathbf{FTL}$ be a set of timelines, $t \in \mathbb{T}$, and θ a scheduling function for $\mathbf{FTL}$. The *prehistory of θ relative to t* is the partial function

$$\theta_{\prec t} : \text{tokens}_U(\mathbf{FTL}) \to \mathbb{T}$$

such that:

$$\theta_{\prec t}(x^i) = \begin{cases} \theta(x^i) & \text{if } \theta(x^i) < t \text{ and } i = 1 \\ \theta(x^i) - \theta(x^{i-1}) & \text{if } \theta(x^i) < t \text{ and } i > 1 \\ \text{undefined} & \text{if } \theta(x^i) \geq t \end{cases}$$

The prehistory $\theta_{\prec t}$ is a partial situation which specifies the durations of the uncontrollable tokens that finish before t according to θ: it is defined only for the uncontrollable tokens x^i such that $\theta(x^i) < t$. When $\theta_{\prec t}(x^i)$ is defined, its value is the (exact) duration of the token x^i in the timelines induced by θ: $\theta_{\prec t}(x^i) = \theta(x^i) - \theta(x^{i-1})$, except when $i = 1$ (x^i is the first token of a timeline), where $\theta_{\prec t}(x^1) = \theta(x^1)$. Consequently, a prehistory defines a partial projection of $\mathbf{FTL}$ fixing the duration of uncontrollable tokens whose end points are less than t according to θ.

Note that, notwithstanding the notation used for prehistories, $\theta_{\prec t}$ is not a (partial) scheduling function, but a partial situation: it assigns values to the durations of uncontrollable tokens.

Example 16 Let $\mathbf{FTL}$ be the set of timelines of example 14 and θ the scheduling function of the same example. If $t \leq 150$, $\theta_{\prec t}$ is undefined everywhere. If $150 < t \leq 220$, $\theta_{\prec t}$ is the function such that $\theta_{\prec t}(y^1) = 150$ (the duration of y^1 in the timeline STL_y) and is undefined elsewhere. If $t > 220$, then $\theta_{\prec t}$ is defined on both the uncontrollable tokens in $\mathbf{FTL}$: $\theta_{\prec t}(y^1) = 150$ and $\theta_{\prec t}(y^2) = 70$.

Definition 22 Let $\Pi = (\mathbf{FTL}, \mathcal{R})$ be a flexible plan for the planning problem $\mathcal{P}$. A *dynamic execution strategy* (DES) for Π is an execution strategy σ for Π such that, for all relevant situations $\omega, \omega' \in \Omega_{\mathbf{FTL}, \mathcal{P}}$ and every controllable token $x^i \in \text{tokens}_C(\mathbf{FTL})$, if $\sigma(\omega) = \theta$, $\sigma(\omega') = \theta'$ and $\theta(x^i) = t$, then

$$\theta_{\prec t} = \theta'_{\prec t} \text{ implies } \theta'(x^i) = \theta(x^i)$$

Example 17 Let us consider the plan $\Pi' = (\mathbf{FTL}', \mathcal{R}')$, where $\mathbf{FTL}'$ contains the timelines

$$\begin{aligned}
FTL_x = x^1 &= (v_1, [20, 30], [20, 30], c), \\
x^2 &= (v_2, [40, 40], [10, 20], c) \\
FTL_y = y^1 &= (v'_1, [10, 15], [10, 15], u), \\
y^2 &= (v'_2, [40, 50], [20, 35], u)
\end{aligned}$$

and $\mathcal{R}' = \{y^1 \text{ end\_before\_start}_{[10,10]} x^2\}$ (i.e., x^2 is constrained to start 10 time units after the end of y^1). Let σ be the execution strategy such that, for every situation ω, $\sigma(\omega)$ is the scheduling function θ such that $\theta(x^1) = \omega(y^1) + 10$, $\theta(x^2) = 40$, $\theta(y^1) = \omega(y^1)$, $\theta(y^2) = \omega(y^1) + \omega(y^2)$. If ω and ω' are two situations, $\sigma(\omega) = \theta$, and $\sigma(\omega') = \theta'$, then $\theta(x^2) = \theta'(x^2)$, so the condition for σ to be a DES has to be checked only for the token x^1. Let $\theta(x^1) = t \in [20, 30]$. Since $\theta(y^1) \in [10, 15]$, $\theta(y^1) < \theta(x^1)$. Therefore, if $\theta_{\prec t} = \theta'_{\prec t}$, then also $\theta(y^1) = \theta'(y^1)$ and, consequently, $\theta'(x^1) = \omega'(y^1) + 10 = \theta'(y^1) + 10 = \theta(y^1) + 10 = \omega(y^1) + 10 = \theta(x^1) = t$. This implies that σ is a dynamic execution strategy.

Finally, the controllability properties considered for STNUs – weak, strong and dynamic controllability – can be defined on flexible plans. In simple terms, if the executor of a weakly controllable plan knows in advance how the uncontrollable events will evolve, i.e., the complete relevant situation ω, it can safely adopt the decisions induced by the scheduling function associated to ω. If the events turn out to be like modeled by ω, such decisions lead to a successful execution of the plan. But in case actual events evolve differently from ω, the executor might be unable to adapt its strategy to the new situation. Therefore, the higher the level of uncertainty in the plan, the higher is the probability to fail while executing it. When, on the contrary, a plan is strongly controllable, its executor is on a safe side: whichever the uncontrollable events turn out to be, it can take the same decisions (posted by a fixed scheduling function) to face the situation and successfully complete the execution of the plan. Unfortunately, few plans are strongly controllable, especially in highly dynamic domains. Finally, when a plan is dynamically controllable, its executor has to monitor what is happening in the world step by step, and decide what to do accordingly. However, it is always sure to be able to adapt its schedules: at each step, whatever happened in the past, there is a decision that can be taken for the next controllable event, in such a way that the plan will at the end be executed successfully. As a consequence, the desired goals can be achieved for any possible turnout of uncontrollable events. Dynamic controllability constitutes a highly desirable property for a flexible plan Π. As a matter of fact, the associated viable DES can be exploited to endow a timeline-based control architecture ensuring robust plan execution (see for instance [12]).

The three controllability properties are formally defined for flexible plans as follows.

Definition 23 Let $\Pi = (\mathbf{FTL}, \mathcal{R})$ be a flexible plan for the planning problem $\mathcal{P}$.

The plan Π is *weakly controllable* if there is a viable execution strategy for Π.

The plan Π is *strongly controllable* if there is a viable execution strategy σ such that, for all relevant situations $\omega, \omega' \in \Omega_{\mathbf{FTL}, \mathcal{P}}$, if $\sigma(\omega) = \theta$ and $\sigma(\omega') = \theta'$, then for every controllable token $x^i \in \text{tokens}_C(\mathbf{FTL})$:

$$\theta(x^i) = \theta'(x^i)$$

The plan Π is *dynamically controllable* if there exists a viable DES for Π.

Example 18 Let us consider the flexible plan $\Pi = (\mathbf{FTL}, \mathcal{R})$ and the viable execution strategy σ of example 15. Since a viable execution strategy exists for Π, it is weakly controllable. However, it is not strongly controllable: the values assigned to token end points by $\sigma(\omega)$, for any execution strategy σ, necessarily depend on the value of $\omega(y^1)$. Nor is Π dynamically controllable. Intuitively, the decision on the end time of x^1 (and consequently of x^2) depend on the end time of y^1, which is in the future. Formally, let us consider, for instance, situations ω and ω', with $\omega(y^1) = 140$ and $\omega'(y^1) = 150$. Let σ be any excution strategy, $\sigma(\omega) = \theta$, and $\sigma(\omega') = \theta'$. In order for σ to be viable, it must be $\theta(x^1) = 110$ and $\theta'(x^1) = 120$. Both prehistories $\theta_{\prec 110}$ and $\theta'_{\prec 110}$ are undefined everywhere, since both $\theta(y^i) > 110$ and $\theta'(y^i) > 110$, for $i = 1, 2$. However $\theta'(x^1) \neq 110$.

As an example of a strongly controllable plan, let us consider $\Pi'' = (\mathbf{FTL}'', \mathcal{R}'')$ with $\mathbf{FTL}''$ containing the two timelines

$$
\begin{aligned}
FTL_x = x^1 &= (v_1, [10, 20], [10, 20], c), \\
x^2 &= (v_2, [50, 50], [30, 40], c) \\
FTL_y = y^1 &= (v'_1, [15, 30], [15, 30], u), \\
y^2 &= (v'_2, [50, 60], [20, 35], u)
\end{aligned}
$$

and $\mathcal{R}'' = \{x^1 \text{ end\_before\_start}_{[0, \infty]} y^2\}$. Let σ be the strategy such that, for every situation ω, $\sigma(\omega)$ is the scheduling function θ such that $\theta(x^1) = 10$, $\theta(x^2) = 50$, $\theta(y^1) = \omega(y^1)$, and $\theta(y^2) = \omega(y^1) + \omega(y^2)$. Whatever is the value of $\omega(y^1)$, $\text{end\_time}(y^1) = \text{start\_time}(y^2) \geq 15$, therefore $x^1 \text{ end\_before\_start}_{[0, \infty]} y^2$ is satisfied by the set of timelines induced by θ, i.e., θ is consistent with Π'' and σ is a viable execution strategy. As a consequence, Π'' is strongly controllable.

Finally, let us consider the plan Π' of example 17. The existence of the viable DES σ implies that Π' is dynamically controllable.

Cialdea Mayer and Orlandini [15] define a semantics of flexible plans in terms of networks of Timed Game Automata (TGA), showing how they can be encoded into such networks. The translation allows one to exploit existing verification tools for TGA in order to check the dynamic controllability property for flexible plans and, possibly, generate a dynamic execution strategy that can be used for robust plan execution.

7 Related Work and Concluding Remarks

This paper proposes the first (up to our knowledge) comprehensive formal account of flexible timelines and plans with uncertainty, combining and extending the work

by Cialdea Mayer et al. [16] and Cialdea Mayer and Orlandini [15].A general semantics is provided for related planning concepts such as domains, goals, problems, constraints and flexible plans, introducing quantitative temporal relations and taking into account the difference between controllable and uncontrollable activities. Execution aspects and controllability issues are also taken into account, and the distinction between controllable and uncontrollable components of the system is modeled (i) introducing the notion of "external" state variable, modeling components that are outside the planner control, and (ii) considering the possibility that the exact duration of a controllable event cannot be foreseen exactly. Finally, such notions are exploited in order to formally define different forms of plan controllability.

The main motivation of this work relies on the lack in the current state of the art of a formal reference framework capturing the modeling features of most of the timeline-based approaches, such as CBTP [5, 18, 20], APSI-TRF [7], IxTeT [25] and the one proposed by Cimatti et al. [17]. They share the view of plans as essentially made up of sets of flexible timelines allowing one to model complex systems as a set of relevant features that must be controlled in time aiming at obtaining some desired behavior. However, with the exception of Cimatti et al., who nevertheless do not explicitly embed temporal flexibility into timelines, the above-mentioned works do not provide a formally grounded definition of the exploited concepts, languages and tools. More generally, they rely on external structures, such as temporal networks (CBTP), decision networks (APSI), or constraint satisfaction problems (IxTeT), to represent constraints among intervals/events. Consequently, the notion of solution plan is associated with the particular implementation of the plan itself. It is worth pointing out that one of the main points of strength of the formalization proposed in this work is specifically connected to this issue: the proposed framework provides a homogeneous and generalized representation of planning domains, timelines and flexible plans, on the basis of which the notion of valid plan is defined independently from the specific representation technology. In this way, controllability issues can be addressed independent of both the exploited plan representation and the solving engine. The importance of this feature is due to the fact that representing a flexible timeline-based plan by means of different representations, such as, e.g., temporal networks, entails a sort of simplification of the associated plan structure, thus causing a loss of information on the causal/temporal "dependencies" among its components.

The expressiveness provided by the synchronization rules defined in the present work aims at capturing the corresponding structures used to state operational constraints in the other considered approaches. The introduction of disjunctions and unambiguous references to valued intervals by means of token variables makes the right-hand side of synchronization rules more expressive and semantically clearer than in other approaches. The formalism proposed by Cimatti et al. [17] to express synchronizations allows neither disjunctions nor token variables, thus preventing to mention different intervals having the same value. A formal comparison with respect to the expressive power cannot be made for the other approaches, because of the lack of a formal counterpart. Thus, our analysis is based on practical considerations and application experiences.

Concerning uncertainty and controllability issues, this work follows and extends the work by Cialdea Mayer and Orlandini [15] considering also planning domains and other associated definitions, as well as enhancing and correcting some flaws in

the definitions of observations and the controllability properties. More generally, this work advances the state of the art in several ways. In fact, the definition of planning domains, in both Apsi-Trf and IxTeT, allows one to model uncertainty only by means of timelines for external variables and not tokens with uncontrollable durations. On the other hand, in Cbtp uncertainty is not considered at all in the description of the planning domain. In fact, the Cbtp framework addresses the controllability problem by relying on the representation of a flexible plan as an STNU. Similarly, Apsi-Trf does not directly addresses controllability issues, but it exploits a model checking approach based on Timed Game Automata (TGA) [11, 29]. In [1], Abdedaim et al. propose a mapping from IxTeT planning problems to TGA reachability problems and the use of a model checking tool for plan synthesis, but they do not explicitly address controllability issues. On the contrary, Cimatti et al. [17] use an explicit annotation of state variable values, partitioning them into controllable and uncontrollable ones, similar to what is done by means of the controllability tagging function of the present work. Synchronization rules may also be annotated as uncontrollable, though the semantics of such annotation is left implicit. Finally, strong controllability is the only property taken into account for plans by Cimatti et al. [17], thus preventing the deployment of least commitment control approaches that are usually more effective while dealing with highly dynamic systems.

As a further remark, it is worth pointing out that, although the main notions defined for STNU and concerning controllability can be transposed into the timeline-based framework presented in this paper (as shown in Section 6), no formal analysis of the relation between the two frameworks has ever been done. As a matter of fact, it would not have been possible, since the timeline-based framework (with temporal flexibility and uncertainty) lacked a formally grounded specification. The contribution of the present work opens the way to addressing such an important issue.

In conclusion, the formalization presented in this work aims at constituting a unified framework where the main timeline-based approaches to P&S can be embedded. Since the focus here is on causal and temporal reasoning, additional interesting features, such as, for instance, the representation of *sharable* and *unsharable* resources modeled in IxTeT, are currently not addressed and they can be considered as possible future work.

As a final remark, it is worth pointing out that the formal framework proposed in this paper allows for the definition of complex planning specifications. In this regard, even though the deployment of P&S applications for autonomous systems in real-world contexts achieved several successes over the last decade, the proposed models and solutions turn out very often to be neither obvious nor immediately acceptable for users. Validating and verifying the correctness and effectiveness of these systems by simple inspection can indeed be a very complex task [4]. In this regard, Knowledge Engineering environments for P&S systems [3, 31, 32, 34] are required to simplify the access to this technology and, then, to provide support for the development of applications to an enlarged community of users (not just "leading edge" specialists). Particularly relevant to the present framework is the exploitation of Timed Game Automata model checking techniques, proposed by Orlandini et al. [31], to integrate verification and validation (V&V) tools and formal methods.

As far as future work is concerned, two important issues to be addressed concern expressivity and complexity. As a matter of fact, being the contribution of the present work the first well-founded comprehensive formalization of flexible timelines with uncertainty, it opens the way to the analysis of the expressive power of the formalism itself and its comparison with analogous ones, such as STNUs, as well as the study of the computational complexity of related planning problems.

On a more practical side, the following issues are currently under investigation: (i) the definition of a specification language compliant with the formal framework and aiming at constituting a reference language for other already existing P&S applications, and (ii) the definition of generalized algorithms and heuristics capable of managing controllability issues while solving P&S problems. The long-term goal is to operationalize all of the above results, contributing to the development of richer software environments and to synthesize a new generation of robust problem-solving applications.

References

1. Abdedaim Y, Asarin E, Gallien M, Ingrand F, Lesire C, Sighireanu M (2007) Planning robust temporal plans: A comparison between CBTP and TGA approaches. In: Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS 2007), pp 2–10
2. Allen JF (1983) Maintaining knowledge about temporal intervals. Communications of the ACM 26(11):832–843
3. Barreiro J, Boyce M, Do M, Frank J, Iatauro M, Kichkaylo T, Morris P, Ong J, Remolina E, Smith T, Smith D (2012) EUROPA: A Platform for AI Planning, Scheduling, Constraint Programming, and Optimization. In: The 4th Int. Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS 2012)
4. Bensalem S, Havelund K, Orlandini A (2014) Verification and validation meet planning and scheduling. International Journal on Software Tools for Technology Transfer 16(1):1–12
5. Bernardini S (2008) Constraint-based temporal planning: Issues in domain modelling and search control. PhD thesis, Università degli Studi di Trento
6. Castillo L, Cortellessa G, Yorke-Smith N (2011) Special issue on scheduling and planning applications: Selected papers from the SPARK workshop series. Computational Intelligence 27(1)
7. Cesta A, Fratini S (2008) The Timeline Representation Framework as a Planning and Scheduling Software Development Environment. In: Proceedings of the 27th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG-08)
8. Cesta A, Oddi A (1996) DDL.1: A Formal Description of a Constraint Representation Language for Physical Domains. In: Ghallab M, Milani A (eds) New Directions in AI Planning, IOS Press, pp 341–352

9. Cesta A, Cortellessa G, Fratini S, Oddi A, Policella N (2007) An Innovative Product for Space Mission Planning: An A Posteriori Evaluation. In: Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling (ICAPS 2007), pp 57–64

10. Cesta A, Cortellessa G, Fratini S, Oddi A (2009) Developing an End-to-End Planning Application from a Timeline Representation Framework. In: Proceedings of the 21st Conference on Innovative Applications of Artificial Intelligence (IAAI-09), pp 66–71

11. Cesta A, Finzi A, Fratini S, Orlandini A, Tronci E (2010) Analyzing flexible timeline-based plans. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010), pp 471–476

12. Cesta A, Cortellessa G, Fratini S, Oddi A (2011) MRSPOCK: Steps in developing an end-to-end space application. Computational Intelligence 27(1):83–102

13. Chien S, Tran D, Rabideau G, Schaffer S, Mandl D, Frye S (2010) Timeline-based space operations scheduling with external constraints. In: Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS 2010), pp 34–41

14. Chien SA, Johnston M, Frank J, Giuliano M, Kavelaars A, Lenzen C, Policella N (2012) A generalized timeline representation, services, and interface for automating space mission operations. In: Proceedings of Space Operations (SpaceOps) 2012

15. Cialdea Mayer M, Orlandini A (2015) An executable semantics of flexible plans in terms of Timed Game Automata. In: Proceedings of the 22nd International Symposium on Temporal Representation and Reasoning (TIME 2015), pp 160–169

16. Cialdea Mayer M, Orlandini A, Umbrico A (2014) A formal account of planning with flexible timelines. In: Proceedings of the 21st International Symposium on Temporal Representation and Reasoning (TIME 2014), pp 37–46

17. Cimatti A, Micheli A, Roveri M (2013) Timelines with temporal uncertainty. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence, pp 195–201

18. Finzi A, Pirri F (2005) Representing flexible temporal behaviors in the Situation Calculus. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05), pp 436–441

19. Frank J (2013) What is a timeline? In: Proceedings of the 4th Workshop on Knowledge Engineering for Planning and Scheduling (KEPS 2013), pp 31–38

20. Frank J, Jónsson A (2003) Constraint-based attribute and interval planning. Constraints 8(4):339–364

21. Fratini S, Pecora F, Cesta A (2008) Unifying Planning and Scheduling as Timelines in a Component-Based Perspective. Archives of Control Sciences 18(2):231–271

22. Hunsberger L (2009) Fixing the semantics for dynamic controllability and providing a more practical characterization of dynamic execution strategies. In: Proceedings of the 16th International Symposium on Temporal Representation and Reasoning (TIME 2009), pp 155–162

23. Hunsberger L (2010) A fast incremental algorithm for managing the execution of dynamically controllable temporal networks. In: Proceedings of the 17th International Symposium on Temporal Representation and Reasoning (TIME

2010), pp 121–128

24. Jonsson A, Morris P, Muscettola N, Rajan K, Smith B (2000) Planning in interplanetary space: Theory and practice. In: Proceedings of the 5th International Conference on Artificial Intelligence Planning Systems (AIPS '00), pp 177–186

25. Laborie P, Ghallab M (1995) IxTeT: an integrated approach for plan generation and scheduling. In: Proceedings of the INRIA/IEEE Symposium on Emerging Technologies and Factory Automation (ETFA'95), pp 485–495

26. Morris P (2014) Dynamic controllability and dispatchability relationships. In: Integration of AI and OR Techniques in Constraint Programming, Proceedings of the 11th International Conference, CPAIOR 2014, Springer, LNCS, vol 8451, pp 464–479

27. Morris PH, Muscettola N, Vidal T (2001) Dynamic control of plans with temporal uncertainty. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001), pp 494–502

28. Muscettola N (1994) HSTS: Integrating planning and scheduling. In: Zweben, M and Fox, MS (ed) Intelligent Scheduling, Morgan Kauffmann, pp 169–212

29. Orlandini A, Finzi A, Cesta A, Fratini S (2011) TGA-based controllers for flexible plan execution. In: KI 2011: Advances in Artificial Intelligence, Proceedings of the 34th Annual German Conference on AI, Springer, LNCS, vol 7006, pp 233–245

30. Orlandini A, Suriano M, Cesta A, Finzi A (2013) Controller synthesis for safety critical planning. In: Proceedings of the IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI 2013), pp 306–313

31. Orlandini A, Bernardi G, Cesta A, Finzi A (2014) Planning meets verification and validation in a knowledge engineering environment. Intelligenza Artificiale 8(1):87–100

32. Simpson RM, Kitchin DE, McCluskey TL (2007) Planning Domain Definition using GIPO. Knowledge Engineering Review 22(2):117–134

33. Umbrico A, Orlandini A, Cialdea Mayer M (2015) Enriching a temporal planner with resources and a hierarchy-based heuristic. In: AIxIA 2015 Advances in Artificial Intelligence, Proceedings of the 14th International Conference of the Italian Association for Artificial Intelligence, Springer, LNCS, vol 9336, pp 410–423

34. Vaquero TS, Silva JR, Tonidandel F, Beck JC (2013) itSIMPLE: Towards an Integrated Design System for Real Planning Applications. Knowledge Engineering Review 28(2):215–230

35. Vidal T, Fargier H (1999) Handling contingency in temporal constraint networks: from consistency to controllabilities. Journal of Experimental and Theoretical Artificial Intelligence 11:23–45