Linköping Electronic Articles in
Computer and Information Science
Vol. 4(1999): nr 11

Fluent Calculus Planning with Continuous Change

Michael Thielscher

Department of Computer Science
Dresden University of Technology
Dresden, Germany

Linköping University Electronic Press
Linköping, Sweden

http://www.ep.liu.se/ea/cis/1999/011/

Published on July 21, 1999 by
Linköping University Electronic Press
581 83 Linköping, Sweden

Linköping Electronic Articles in
Computer and Information Science
ISSN 1401-9841
Series editor: Erik Sandewall

Typeset by the author using LaTeX
Formatted using étendu *style*

Recommended citation:
< Author > . < Title > . Linköping Electronic Articles in Computer and Information Science, Vol. 4(1999): nr 11. http://www.ep.liu.se/ea/cis/1999/011/. *July 21, 1999.*

This URL will also contain a link to the author's home page.

Abstract

Green's classical definition of planning by deduction has recently been generalized to hybrid domains, which involve both discrete and continuous change. We show that this generalization is unsuited for planning under incomplete information. An improved theory is developed on the basis of the novel concept of a situation tree with trajectories and using the predicate formalism of the Fluent Calculus.

Some of the results in this paper have been preliminarily reported in [17].

1 Introduction

Research in Cognitive Robotics aims at explaining and modeling intelligent acting in a dynamic world. The classical Frame Problem is the most fundamental theoretical challenge towards this end if intelligent behavior is understood as resulting from correct reasoning and if it cannot reasonably be assumed that an agent possesses complete knowledge of world states. Actually, the Frame Problem comes with two facets: a *representational* one, which concerns the effort needed to specify non-effects of actions, and an *inferential* one, which concerns the effort needed to actually infer these non-effects. The Fluent Calculus, which roots in the logic programming formalism of [3], provides an axiomatization strategy that particularly aims at the second aspect. A new, expressive version of this calculus has recently been developed around the concept of so-called state update axioms [18]. In this paper we extend this formalism to reasoning about actions and planning in hybrid domains, which involve both discrete and continuous change.

The first step towards this end is the insight that the fluents themselves, that is, the atomic components of descriptions of world states, can be hybrid. Generally, a fluent is assumed to be stable in between the occurrence of two consecutive actions, and yet a fluent may internally represent an arbitrarily complex, continuous process [2]. A physical object x moving through an n-dimensional space with constant spatial velocity v, for instance, can be represented by a process $Movement(x, p_0, v, t_0)$, where the parameter p_0 denotes the location of the object at the time t_0 when the motion was initiated. Although it describes a particular continuous change, the 'process fluent' $Movement$ itself remains unchanged unless some action causes a discontinuity.[1] World states in hybrid systems can thus be modeled as collections of both ordinary fluents and process fluents, which may be initiated and terminated by actions.

In a world full of ongoing processes, however, an agent executing a plan is not the only source of discontinuities. Also the laws of physics frequently imply state transitions. If, for instance, two moving objects eventually meet, then an 'autonomous' update of the system state happens. In a recent generalization proposed in [11] of Green's classical definition of planning by deduction [1], it is shown that these exogenous (or: *natural*) actions can be specified just like deliberative actions by preconditions and effects. Using the Situation Calculus paradigm [7], a plan in this approach is defined as a sequence of action performances $Do(a_n, Do(a_{n-1}, \ldots, Do(a_1, S_0)\ldots))$, starting in some initial situation S_0 and involving both deliberative actions by the agent and all natural actions that happen in the meantime. For example, the term

$$Do(\mathit{Falls\_into}(\mathit{Object}, \mathit{Container}, 0{:}10), Do(\mathit{Send}(\mathit{Container}, 0{:}03), S_0))$$

may denote the plan to send a $Container$ on a conveyor belt at time 0:03 right on time such that at time 0:10 the container will automatically catch an $Object$ which travels on a second belt that ends somewhere above the first one (see also Fig. 1). Here, $Send(Container, 0{:}03)$ is a deliberative, $\mathit{Falls\_into}(\mathit{Object}, \mathit{Container}, 0{:}10)$ a natural action. In this setting, it is of course not for the agent to freely choose which natural actions to insert into or leave out from her plan. That is to say, only those plans are to be considered in which all expected natural actions do indeed happen as nature dictates [12]. This inevitable condition is met in [11] by distinguishing so-called

[1] Other examples of process fluents could be: acceleration, oscillation, dilatation and contraction, etc.

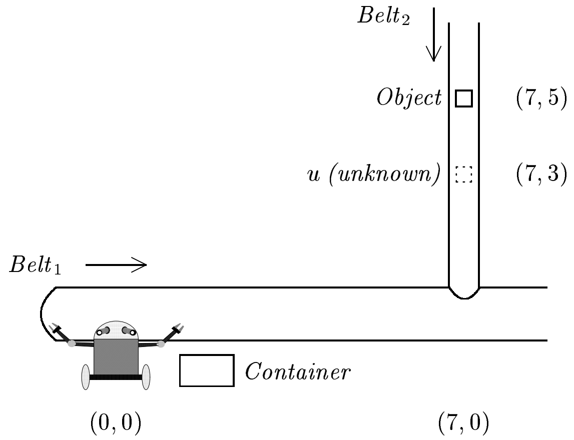

Figure 1: A detail of a production line: Conveyor $Belt_1$ travels with constant velocity to the right, conveyor $Belt_2$ with constant velocity towards $Belt_1$. Incoming objects that reach the end of $Belt_2$ fall automatically onto $Belt_1$.

legal plans where it is required for each situation $Do(a_i, \ldots, Do(a_1, S_0) \ldots)$ that no natural action is expected which occurs before the (deliberative or natural) action a_{i+1} $(0 \leq i \leq n)$. In other words, prior to performing a deliberative action in a certain situation, the latter needs to be updated by all natural actions, in the right order, which happen before.

However, we will show that this approach is unsuited for planning under incomplete information. In a nutshell, the problem is that incomplete information may cause uncertainty about whether some natural action will happen or not. A planning problem may then admit no provably legal plan even if a simple sequence of deliberative actions will achieve the goal in any case. Coming from this insight, we develop a theory of Fluent Calculus planning with continuous change in which natural actions are uncoupled from situations. To this end, we introduce the concept of a situation tree with trajectories, where situations are sequences of *deliberative* actions and where the world state at any situation may evolve autonomously according to *natural* actions. On this basis we obtain a better generalization of the classical definition of planning by deduction, which is applicable to planning problems with incomplete information about world states.

The remainder of the paper is organized as follows. In the following section, we reveal the problem with the definition of [11] when it comes to planning under incomplete knowledge of world states. In Section 3 we extend the basic Fluent Calculus to continuous change and natural actions. Our theory assumes deterministic worlds where actions or events cannot occur concurrently. In Section 4 we show that our approach is well-suited for planning under incomplete information, and in Section 5 we show how our theory incidentally also enables solutions to Zeno's paradox, which is yet another, more theoretical problem arising in the approach of [11]. The paper concludes with a discussion of our results in Section 6.

2 Planning under incomplete information

Consider the planning problem sketched in Fig. 1. Initially, at time 0:00, the robot has possession of a *Container*, which it may put onto conveyor $Belt_1$ at global position $(0,0)$ at any time. At the moment nothing is on $Belt_1$. An *Object* is known to travel on conveyor $Belt_2$, initially located at position $(7,5)$. There is one piece of incomplete information, namely, it is unknown whether or not another, unspecific object u is on $Belt_2$ at position $(7,3)$. The constant velocity of $Belt_1$ shall be 1 unit per second, and that of $Belt_2$ shall be 0.5 unit per second. The robot's goal is to have the *Container* and the *Object* traveling together on $Belt_1$.

Simple arithmetics tells us that the straightforward solution to this planning problem is for the robot to send the *Container* along $Belt_1$ at time 0:03. According to the specification, the *Container* will then arrive at position $(7,0)$ at time 0:10, which happens to be exactly the time at which the *Object* reaches the end of $Belt_2$ and will fall down. This solution obviously works no matter whether $Belt_2$ initially carries this other object u, of which we can say that if it were present, then it would fall onto $Belt_1$ at time 0:06. But this is irrelevant as far as the solution to the planning problem is concerned.

Formalized in the approach of [11], however, this simple planning problem has no provably correct solution. To see why, let $Send(x,t)$ denote the deliberative action of putting x onto $Belt_1$ at time t, and let $Falls(x,t)$ denote the natural action of x falling down from $Belt_2$.[2] Let the fluent $Has(x,s)$ denote that the robot has possession of x in situation s, and, as sketched above, let the fluent $Movement(x,(p_x,p_y),(v_x,v_y),t,s)$ denote that x at time t started to move away from position (p_x,p_y) with constant spatial velocity (v_x,v_y) in situation s. Given this signature it is straightforward to devise two suitable successor state axioms for Has and $Movement$, resp.:[3]

$$\begin{array}{l} Poss(a,s) \supset \\ \quad Has(x,Do(a,s)) \equiv Has(x,s) \wedge \neg\exists t.\, a = Send(x,t) \end{array}$$

$$\begin{array}{l} Poss(a,s) \supset \\ \quad Movement(x,p,v,t,Do(a,s)) \equiv \\ \qquad\quad a = Send(x,t) \wedge p = (0,0) \wedge v = Vel(Belt_1) \\ \qquad \vee\ a = Falls(x,t) \wedge p = (7,0) \wedge v = Vel(Belt_1) \\ \qquad \vee\ Movement(x,p,v,t,s) \wedge \neg\exists t'.\, a = Falls(x,t') \end{array}$$

where $Poss(a,s)$ means that action a is possible in situation s. The appropriate action precondition axioms for $Send$ and $Falls$, resp., are:

$$Poss(Send(x,t),s) \equiv Has(x,s)$$

$$\begin{array}{l} Poss(Falls(x,t),s) \equiv \exists p,t_0\, [\, Movement(x,p,Vel(Belt_2),t_0,s) \wedge \\ \qquad\qquad\qquad\qquad\qquad p + Vel(Belt_2)\cdot(t-t_0) = (7,0)\,] \end{array}$$

Let $Axioms$ consist of these four axioms plus a suitable axiomatization of

[2] A word on the notation: Predicate and function symbols, including constants, start with a capital letter whereas variables are in lower case, sometimes with sub- or superscripts. Free variables in formulas are assumed universally quantified.

[3] For the sake of simplicity we assume a non-concurrent world here; see also Section 6.

the information about the specific planning problem,

$$Vel(Belt_1) = (1,0) \wedge Vel(Belt_2) = (0,-0.5)$$

$$\begin{array}{l} Has(Container, S_0) \wedge \\ Movement(Object, (7,5), Vel(Belt_2), 0{:}00, S_0) \wedge \\ \exists u\, [\, Movement(x,p,v,t,S_0) \supset \\ \qquad x = Object \wedge p = (7,5) \wedge v = Vel(Belt_2) \wedge t = 0{:}00 \\ \qquad \vee\; x = u \wedge p = (7,3) \wedge v = Vel(Belt_2) \wedge t = 0{:}00\,] \end{array}$$

along with the standard axioms of [11], including a suitable definition of $Natural(a)$ indicating that a is a natural action,

$$Natural(a) \equiv \exists x,t.\, a = Falls(x,t)$$

and a definition of $Legal(s)$ indicating that s is a legal situation as outlined in the introduction:

$$\begin{array}{l} Legal(S_0) \wedge \\ [\, Legal(Do(a,s)) \equiv \\ \quad Legal(s) \wedge Poss(a,s) \wedge Start(s) \leq Time(a) \wedge \\ \quad \forall a' (Natural(a') \wedge Poss(a',s) \supset a = a' \vee Time(a) < Time(a'))\,] \end{array}$$

where $Time(a)$ denotes the execution time of action a,

$$Time(Send(x,t)) = t \;\wedge\; Time(Falls(x,t)) = t$$

and $Start(s)$ denotes the time at which situation s arises,

$$Start(S_0) = 0 \;\wedge\; Start(Do(a,s)) = Time(a)$$

Finally, let $G(s)$ be a formula stating that in situation s both $Container$ and $Object$ travel on $Belt_1$ at the very same position:

$$\begin{array}{l} \exists p_1, p_2, t_1, t_2\, [\, Movement(Container, p_1, Vel(Belt_1), t_1, s) \wedge \\ \qquad Movement(Object, p_2, Vel(Belt_1), t_2, s) \wedge \\ \qquad p_1 + Vel(Belt_1) \cdot (t_2 - t_1) = p_2\,] \end{array}$$

Now, [11] gives a Situation Calculus-based definition of deductive planning with continuous processes and natural actions, by which is generalized Green's classical definition [1]. Applied to our example, the planning problem is to find a plan σ such that $Axioms \models Legal(\sigma) \wedge G(\sigma)$. However, no such σ exists. The only two reasonable candidates σ_1 and σ_2, resp., are,

$$Do(Falls(Object, 0{:}10), Send(Container, 0{:}03), S_0))$$
$$Do(Falls(Object, 0{:}10), Do(Falls(u, 0{:}06), Do(Send(Container, 0{:}03), S_0)))$$

But $Axioms$ do admit a model in which $\neg Legal(\sigma_1)$ holds, namely, where initially the unknown object is present, in which case there is an action a, viz. $Falls(u, 0{:}06)$, such that

$$Natural(a) \wedge Poss(a, Do(Send(Container, 0{:}03), S_0)) \wedge Time(a) < 0{:}10$$

Likewise, $Axioms$ do admit a model in which $\neg Legal(\sigma_2)$ holds, namely, where initially the unknown object is not present, in which case we have

$$\neg \exists u.\, Poss(Falls(u, 0{:}06), Do(Send(Container, 0{:}03), S_0))$$

Hence neither of the two candidates provides a solution to our planning problem, nor does any other plan. This observation indicates that the above definition of deductive planning is unsuited for planning under incomplete information.[4]

[4] The reader may wonder if this drawback cannot be overcome by adopting from [6]

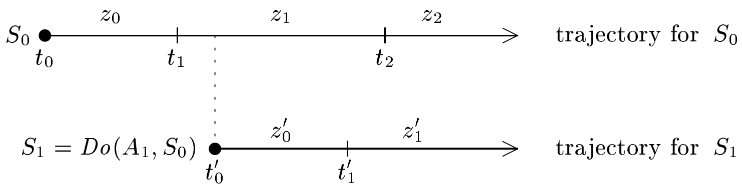

Figure 2: Each situation has its own trajectory, which describes how the state evolves according to the expected natural actions. For instance, in the example shown here, at the time t'_0 when the deliberative action A_1 is performed in situation S_0, the world is no longer in the initial state z_0 due to a natural action happening at time $t_1 < t'_0$. The effect of A_1 is to transform state z_1 into z'_0, which thus becomes the initial state of the trajectory for situation $Do(A_1, S_0)$. (For a concrete example see Fig. 3 later in this paper.)

3 The Fluent Calculus for Natural Actions and Continuous Change

For an easy understanding of the following, some background knowledge of the simple Fluent Calculus is helpful. The ETAI reference article [16], Sections 1 to 4.1, provides an ideal introduction to this end.

Prior to diving into the formal details of how to specify hybrid systems with the Fluent Calculus, let us give some intuition about how we are going to solve the problem of non-existence of plans which has been illustrated in Section 2. The crucial notion is that of a *situation tree with trajectories*.

The natural actions are uncoupled from the situation term. A plan thus becomes what it used to be in the classical definition, namely, a sequence of deliberative actions to be performed by the planning agent. An implication of this view is that natural actions cause state transitions without effecting a new situation. To this end, each situation has a *trajectory*. A trajectory is a sequence of states. The world starts in a particular state whenever a new situation arises. This state shall be the first one on the situation's trajectory. The further evolution of the trajectory is then determined by the natural actions that are expected to happen. The performance of a deliberative action, on the other hand, brings about a new situation with its own trajectory; see Fig. 2.

The Fluent Calculus for domains with natural actions and continuous change is an order-sorted second order language with equality which includes the following:

- Sorts:

  $$\mathit{fluent},\ \mathit{state},\ \mathit{action},\ \mathit{sit},\ \mathit{timepoint}$$

  where *fluent* is a sub-sort of *state*.

the generalized notion of a *conditional* plan, by which can be specified a plan something like this: Do $Send(Container, 0{:}03)$; if the unknown object is not present, then do $Falls(Object, 0{:}10)$, else do $Falls(u, 0{:}06)$ followed by $Falls(Object, 0{:}10)$. But such plans are executable only in case all if-conditions are known to the agent in the situations where the corresponding branching happens. Hence, our example planning problem would still be unsolvable if the robot had no possibility of checking whether the unknown object is present. (Apart from this, commanding the robot to drive up to $Belt_2$ to do the checking would not constitute the expected efficient solution, anyway.)

- Functions:

$$
\begin{aligned}
\emptyset &: \mathit{state} \\
\circ &: \mathit{state} \times \mathit{state} \mapsto \mathit{state} \\
\mathit{State} &: \mathit{sit} \mapsto \mathit{state} \\
\mathit{Do} &: \mathit{action} \times \mathit{sit} \mapsto \mathit{sit} \\
\mathit{Time} &: \mathit{action} \mapsto \mathit{timepoint} \\
\mathit{StartTime} &: \mathit{timepoint} \mapsto \mathit{fluent} \\
\mathit{Succ} &: \mathit{action} \times \mathit{state} \mapsto \mathit{state}
\end{aligned}
$$

- Predicates:

$$
\begin{aligned}
\mathit{Poss} &: \mathit{action} \times \mathit{sit} \\
\mathit{Natural} &: \mathit{action} \\
\mathit{Expect} &: \mathit{action} \times \mathit{state} \\
\mathit{Trajectory} &: \mathit{state} \times \mathit{state}
\end{aligned}
$$

Throughout the paper, we let the sort *timepoint* be the real numbers. We will use the arithmetic functions $+, -, \cdot$ and relations $<, \leq$ and rely on their standard interpretation. Variables of the sort *timepoint* are denoted by the letter t.

We will frequently use the following abbreviation, which is similar to the *Holds* macro of the simple Fluent Calculus but refers directly to states and not to situations:

$$\mathit{HoldsIn}(f, z) \stackrel{\text{def}}{=} \exists z'.\, z = f \circ z'$$

The world state at each situation is required to include the special fluent $\mathit{StartTime}(t)$, denoting the state's starting time t; the start time must be unique:

$$
\begin{aligned}
&\exists t.\, \mathit{Holds}(\mathit{StartTime}(t), s) \wedge \\
&[\,\mathit{Holds}(\mathit{StartTime}(t_1), s) \wedge \mathit{Holds}(\mathit{StartTime}(t_2), s) \supset t_1 = t_2\,]
\end{aligned}
\tag{1}
$$

As in [11], all deliberative and natural actions have a timepoint as argument, which denotes the time of execution or occurrence. We assume, for the sake of uniformity, that this argument be the rightmost one so that action terms are of the form $\alpha(\vec{x}, t)$. The function $\mathit{Time}(a)$ maps actions to their execution time, that is, $\mathit{Time}(\alpha(\vec{x}, t)) = t$.

The axiomatization of a natural action $\alpha(\vec{x}, t)$ consists of two parts, the precondition and the effect. The former is specified by a definitional formula $\mathit{Expect}(\alpha(\vec{x}, t), z) \equiv \Phi_\alpha(\vec{x}, t, z)$ where Φ_α is a first-order formula whose free variables are among $\vec{x}, t, z$ and which includes no term of sort *sit*. Formula Φ_α defines the conditions under which action $\alpha(\vec{x}, t)$ is expected to automatically happen in state z, provided no other actions occur beforehand; for an example see formula (9) in Section 4. (Predicate *Expect* plays the same role for natural actions as *Poss* does for deliberative actions.) The effect of a natural action $\alpha(\vec{x}, t)$ on a state z is specified by an update axiom of this form:

$$
\begin{aligned}
&\mathit{Expect}(\alpha(\vec{x}, t), z) \wedge \Delta(\vec{x}, t, z) \supset \\
&\quad (\exists)\, \mathit{Succ}(\alpha(\vec{x}, t), z) \circ \vartheta^- \circ \mathit{StartTime}(t_0) = z \circ \vartheta^+ \circ \mathit{StartTime}(t)
\end{aligned}
$$

where Δ is a first-order formula whose free variables are among $\vec{x}, t, z$ and which contains no term of sort *sit*. Formula Δ defines the condition under which α causes the specified update from state z to its successor state, which is denoted by $\mathit{Succ}(\alpha, z)$. The terms $\vartheta^+ \circ \mathit{StartTime}(t_0)$ and $\vartheta^- \circ \mathit{StartTime}(t)$ are the negative and positive, resp., effects of the action;

for an example see formula (10) in Section 4. (These formulas correspond to state update axioms for deliberative actions.)

A sequence of successor states is called a trajectory if it is always the earliest natural action which leads to the successor state. The predicate $\mathit{Trajectory}(z, z')$ indicates that state z' eventually occurs on the trajectory starting with state z:

$$\begin{array}{l} \mathit{Trajectory}(z, z) \wedge \\ [\,\mathit{Trajectory}(z, z') \wedge \mathit{NextNatAction}(a, z') \supset \mathit{Trajectory}(z, \mathit{Succ}(a, z'))] \end{array} \tag{2}$$

where we use the following abbreviation:

$$\begin{array}{l} \mathit{NextNatAction}(a, z) \stackrel{\text{def}}{=} \\ \quad \mathit{Natural}(a) \wedge \mathit{Expect}(a, z) \wedge \\ \quad \forall a' \,[\, \mathit{Natural}(a') \wedge \mathit{Expect}(a', z) \wedge \mathit{Time}(a') \leq \mathit{Time}(a) \;\supset\; a = a' \,] \end{array}$$

To this end, each domain specification includes an axiom $\mathit{Natural}(a) \equiv \Pi(a)$ where Π is a first-order formula with free variable a and no term of sort sit. Formula Π defines the conditions under which a is a relevant natural action; for an example see formula (8) in Section 4.

Notice that foundational axiom (2) allows more states on the trajectory apart from those that are reachable by a finite sequence of natural actions. This feature is necessary to overcome Zeno's paradox, as will be discussed in Section 5. On the other hand, we need to make sure that a trajectory does not include arbitrary other states. This will be done with the help of a macro to be introduced next.

Omitting natural actions from situations implies that the state $\mathit{State}(s)$, which is initially associated with a situation s, may evolve as time goes by (recall Fig. 2). We therefore introduce the macro $\mathit{ActualState}(s, t, z)$ to indicate that z is the actual state in situation s at time t:

$$\begin{array}{l} \mathit{ActualState}(s, t, z) \stackrel{\text{def}}{=} \\ \quad \forall t_0 \,[\, \mathit{HoldsIn}(\mathit{StartTime}(t_0), z) \supset \\ \qquad\qquad \mathit{Trajectory}(\mathit{State}(s), z) \wedge t_0 \leq t \wedge \\ \qquad\qquad \neg \exists a (\mathit{Natural}(a) \wedge \mathit{Expect}(a, z) \wedge \mathit{Time}(a) \leq t) \,] \end{array}$$

Put in words, the actual state z in situation s at time t lies on the trajectory of s and started at time $t_0 \leq t$, and future natural actions expected in z will happen later. Using this abbreviation, the following foundational axiom restricts trajectories to linear sequences of states by stipulating that there be a unique actual state at any time:

$$\mathit{ActualState}(s, t, z) \wedge \mathit{ActualState}(s, t, z') \;\supset\; z = z' \tag{3}$$

Deliberative actions are specified as usual by means of precondition axioms and state update axioms. Of course, these specifications need to refer to the actual state of the world at the time the action is performed. To this end, the *Holds* expression is furnished with an additional argument indicating the time at which a fluent is supposed to hold in a situation:

$$\mathit{Holds}(f, s, t) \stackrel{\text{def}}{=} \forall z \,[\, \mathit{ActualState}(s, t, z) \;\supset\; \exists z' .\, z = f \circ z' \,]$$

Finally, for convenience we supply the abbreviations $\mathit{After}(t, s)$ and $\mathit{After}(t, z)$ indicating that t denotes a timepoint later than the time at which situation s and state z, resp., came into being, that is,

$$\begin{array}{l} \mathit{After}(t, s) \stackrel{\text{def}}{=} \forall t_0 \,[\, \mathit{HoldsIn}(\mathit{StartTime}(t_0), \mathit{State}(s)) \supset t > t_0 \,] \\ \mathit{After}(t, z) \stackrel{\text{def}}{=} \forall t_0 \,[\, \mathit{HoldsIn}(\mathit{StartTime}(t_0), z) \supset t > t_0 \,] \end{array}$$

This completes the general description of a strategy of axiomatizing, by means of the Fluent Calculus, domains which involve natural actions and continuous change.

Definition 1 Consider a Fluent Calculus signature for natural actions and continuous change as defined above. A Fluent Calculus domain specification is a set of formulas including the following.

1. Precondition axioms for all deliberative actions α, which take the form,
$$Poss(\alpha(\vec{x},t),s) \equiv \Phi_\alpha(\vec{x},t,s)$$
where Φ_α is a first-order formula with s being the only term of sort *sit*.

2. State update axioms for all deliberative actions α, which take the form,
$$\begin{array}{l} Poss(\alpha(\vec{x},t),s) \supset \\ \quad ActualState(s,t,z) \wedge \Delta(\vec{x},t,z) \supset \\ \qquad (\exists)\, State(Do(\alpha(\vec{x},t),s)) \circ \vartheta^- \circ StartTime(t_0) = \\ \qquad\qquad\qquad\qquad z \circ \vartheta^+ \circ StartTime(t) \end{array}$$
where Δ is a first-order formula without a term of sort *sit*, and ϑ^+, ϑ^- are terms of sort *State* .

3. Precondition axioms for all natural actions α, which take the form,
$$Expect(\alpha(\vec{x},t),z) \equiv \Phi_\alpha(\vec{x},t,z)$$
where Φ_α is a first-order formula without a term of sort *sit*.

4. State update axioms for all natural actions α, which take the form,
$$\begin{array}{l} Expect(\alpha(\vec{x},t),z) \wedge \Delta(\vec{x},t,z) \supset \\ \quad (\exists)\, Succ(\alpha(\vec{x},t),z) \circ \vartheta^- \circ StartTime(t_0) = z \circ \vartheta^+ \circ StartTime(t) \end{array}$$
where Δ is a first-order formula without a term of sort *sit*, and ϑ^+, ϑ^- are terms of sort *State*.

5. An axiom $Natural(a) \equiv \Pi(a)$ where Π is a first-order formula without a term of sort *sit*.

6. The foundational axioms of the basic Fluent Calculus, including a suitable domain-dependent set *EUNA* (see [16]).

7. The foundational axioms of this section, viz. (1)–(3).

■

Consider a set of formulas *Axioms* containing a domain specification and some background knowledge, in particular of the initial situation S_0. Furthermore, let $G(s)$ be a formula stipulating that the planning goal is achieved in situation s. Then a planning problem is defined as the problem of finding a term $\sigma = Do(a_n, \ldots, Do(a_1, S_0)\ldots)$ $(n \geq 0)$ such that
$$Axioms \models POSS(\sigma) \wedge G(\sigma)$$
where $POSS(\sigma) \stackrel{\text{def}}{=} Poss(a_1, S_0) \wedge \ldots \wedge Poss(a_n, Do(a_{n-1}, \ldots, S_0)\ldots)$. In this way we have applied Green's classical definition of deductive planning without alteration to planning with continuous change.

4 Planning under incomplete information revisited

In this section we recast the planning problem of Section 2 into the Fluent Calculus to show that our theory is well-suited for planning under incomplete information. Let us first encode the background conditions of the scenario, that is, the spatial velocity of the two conveyor belts:

$$Vel(Belt_1) = (1,0) \wedge Vel(Belt_2) = (0,-0.5) \tag{4}$$

As in Section 2 we use the fluents $Has(x)$ and $Movement(x, p, v, t_0)$. By the following axiom we specify the incomplete information about the initial situation of the planning problem (c.f. Fig. 1):

$$\begin{aligned}
\exists z\, [\; & State(S_0) = StartTime(0{:}00) \circ Has(Container) \circ \\
& \qquad Movement(Object, (7,5), Vel(Belt_2), 0{:}00) \circ z \\
& \wedge \exists u (\forall x, p, v, t.\, HoldsIn(Movement(x,p,v,t), z) \supset \\
& \qquad x = u \wedge p = (7,3) \wedge v = Vel(Belt_2) \wedge t = 0{:}00)\,]
\end{aligned} \tag{5}$$

The only deliberative action, denoted by $Send(x, t)$, is that of the robot putting something onto conveyor $Belt_1$ at global position $(7,0)$. The preconditions for this action are that it is performed after the arising of the situation and that the agent has possession of x. Formally,

$$Poss(Send(x,t), s) \equiv Holds(Has(x), s, t) \wedge After(t, s) \tag{6}$$

The effect of the action is to initiate a continuous movement in the direction and with the velocity of the first belt. Furthermore, the agent loses possession of the object, and also the starting time changes:

$$\begin{aligned}
& Poss(Send(x,t), s) \supset \\
& \quad ActualState(s,t,z) \supset \\
& \quad\quad \exists t_0\, [\, State(Do(Send(x,t), s)) \circ Has(x) \circ StartTime(t_0) \\
& \quad\quad\quad = z \circ Movement(x, (0,0), Vel(Belt_1), t) \circ StartTime(t)\,]
\end{aligned} \tag{7}$$

In addition to the deliberative action of setting objects into motion, we consider the one natural action of objects x falling from $Belt_2$ onto $Belt_1$, denoted by $Falls(x, t)$. This being the only natural action relevant to our scenario, we assert the following:

$$Natural(a) \equiv \exists x, t.\ a = Falls(x,t) \tag{8}$$

This action is expected at any time t when an x reaches the end of $Belt_2$:

$$\begin{aligned}
& Expect(Falls(x,t), z) \equiv \\
& \quad \exists p, t_0\, [\, HoldsIn(Movement(x, p, Vel(Belt_2), t_0), z) \wedge \\
& \quad\quad p + Vel(Belt_2) \cdot (t - t_0) = (7,0)\,] \\
& \quad \wedge After(t, z)
\end{aligned} \tag{9}$$

The effect of this natural action is to terminate the movement of x on $Belt_2$ and to initiate a corresponding movement on $Belt_1$:

$$\begin{aligned}
& Expect(Falls(x,t), z) \supset \\
& \quad \exists t_0, t_0', p, v. \\
& \quad\quad Succ(Falls(x,t), z) \circ Movement(x, p, v, t_0') \circ StartTime(t_0) \\
& \quad\quad\quad = z \circ Movement(x, (7,0), Vel(Belt_1), t) \circ StartTime(t)
\end{aligned} \tag{10}$$

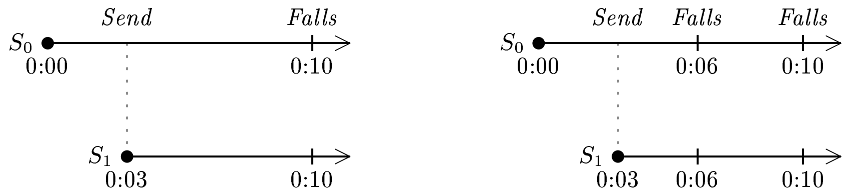

Figure 3: A detail of the trajectory structure in two models for our axiomatization $Axioms_{Belts}$. In the model sketched on the left-hand side the unknown object is not present and, hence, the only natural action that happens in the two depicted trajectories is $Falls(Object, 0{:}10)$. In the model sketched on the right-hand side the additional natural action $Falls(u, 0{:}06)$ happens in the two trajectories for S_0 and S_1. In both models the $Container$ and the $Object$ travel together on $Belt_1$ in all actual states after time $t = 0{:}10$ in situation $S_1 = Do(Send(Container, 0{:}03), S_0)$. The latter thus constitutes a solution to our planning problem.

This completes the axiomatization of the Conveyor Belt domain in the Fluent Calculus. Let $Axioms_{Belts}$ be the union of all axioms of this and the previous section plus the foundational axioms of the simple Fluent Calculus. Let the auxiliary expression $Position(x, p, s, t)$ denote that the position of x in situation s at time t is p:

$$\begin{array}{l} Position(x, p, s, t) \stackrel{\text{def}}{=} \\ \quad Holds(Has(x), s, t) \wedge p = (0, 0) \vee \\ \quad \exists p_0, v, t_0\, [\, Holds(Movement(x, p_0, v, t_0), s, t) \wedge p = p_0 + v \cdot (t - t_0)\,] \end{array}$$

We can then formalize our initial planning problem as follows: Does there exist a situation σ such that $Axioms_{Belts}$ entails

$$\exists p, t\, [\, Position(Container, p, \sigma, t) \wedge Position(Object, p, \sigma, t)\,] \tag{11}$$

Having uncoupled natural actions from plans, the planning problem is no longer rendered unsolvable by the uncertain occurrence of a natural action, which in turn is caused by the incompleteness of the knowledge about the initial situation. The following result shows that our axiomatization indeed entails the expected solution.

Theorem 2 *Let* $\sigma = Do(Send(Container, 0{:}03), S_0)$*, then*

$$Axioms_{Belts} \models (11) \tag{12}$$

Proof: For a sketch of the following formal proof see Fig. 3.
From (5), (9), and (4) it follows that

$$\begin{array}{l} Expect(Falls(Object, 0{:}10), State(S_0)) \wedge \\ \exists u\, [\, Expect(Falls(x, t), State(S_0)) \equiv x = Object \wedge t = 0{:}10 \vee \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad x = u \wedge t = 0{:}06\,] \end{array}$$

Hence, foundational axiom (2) along with (8) implies that

$$ActualState(S_0, 0{:}03, State(S_0)) \tag{13}$$

Thus, $Poss(Send(Container, 0{:}03), S_0)$ according to (6) and (5). By (7), (5), and (13) we then have, given that $\sigma = Do(Send(Container, 0{:}03), S_0)$,

$$\begin{aligned}\exists z\,[\; & State(\sigma) = Movement(Object, (7,5), Vel(Belt_2), 0{:}00) \circ z \circ \\ & \quad Movement(Container, (0,0), Vel(Belt_1), 0{:}03) \circ \\ & \quad StartTime(0{:}03) \\ & \wedge \forall x,p,v,t\,(HoldsIn(Movement(x,p,v,t), z) \supset \\ & \qquad x = U \wedge p = (7,3) \wedge v = Vel(Belt_2) \wedge t = 0{:}00)\,]\end{aligned} \tag{14}$$

for some U. From this and (9) in conjunction with (4) it follows that

$$\begin{aligned}& Expect(Falls(Object, 0{:}10), State(\sigma)) \wedge \\ & [\,Expect(Falls(x,t), State(\sigma)) \equiv x = Object \wedge t = 0{:}10 \vee \\ & \qquad x = U \wedge t = 0{:}06\,]\end{aligned} \tag{15}$$

Hence, by (8),

$$\begin{aligned}& NextNatAction(Falls(Object, 0{:}10), State(\sigma)) \vee \\ & NextNatAction(Falls(U, 0{:}06), State(\sigma))\end{aligned} \tag{16}$$

Thus, from (10), (14), and the first conjunct of (15) it follows that

$$\begin{aligned}\exists z\,[\; & Succ(Falls(Object, 0{:}10), State(\sigma)) = \\ & z \circ Movement(Container, (0,0), Vel(Belt_1), 0{:}03) \circ \\ & Movement(Object, (7,0), Vel(Belt_1), 0{:}10) \circ StartTime(0{:}10)\,]\end{aligned} \tag{17}$$

(as sketched on the left hand side of Fig. 3).

On the other hand, (10) and (14) also imply,

$$\begin{aligned}& Expect(Falls(U, 0{:}06), State(\sigma)) \supset \\ & \quad \exists z\,[\,Succ(Falls(U, 0{:}06), State(\sigma)) = \\ & \qquad Movement(Object, (7,5), Vel(Belt_2), 0{:}00) \circ \\ & \qquad z \circ Movement(Container, (0,0), Vel(Belt_1), 0{:}03) \circ \\ & \qquad Movement(U, (7,0), Vel(Belt_1), 0{:}06) \circ StartTime(0{:}06) \\ & \quad \wedge \neg\exists x,p,v,t.\, HoldsIn(Movement(x,p,v,t), z)\,]\end{aligned} \tag{18}$$

From (9) and (4) it then follows that

$$\begin{aligned}& Expect(Falls(U, 0{:}06), State(\sigma)) \supset \\ & \quad Expect(Falls(x,t), Succ(Falls(U, 0{:}06), State(\sigma))) \equiv \\ & \qquad x = Object \wedge t = 0{:}10\end{aligned}$$

Hence,

$$\begin{aligned}& Expect(Falls(U, 0{:}06), State(\sigma)) \supset \\ & \quad NextNatAction(Falls(Object, 0{:}10), Succ(Falls(U, 0{:}06), State(\sigma)))\end{aligned} \tag{19}$$

Thus, from (18) and (10) it follows that

$$\begin{aligned}& Expect(Falls(U, 0{:}06), State(\sigma)) \supset \\ & \quad \exists z\,[\,Succ(Falls(Object, 0{:}10), Succ(Falls(Object, 0{:}06), State(\sigma))) = \\ & \qquad z \circ Movement(Container, (0,0), Vel(Belt_1), 0{:}03) \circ \\ & \qquad Movement(U, (7,0), Vel(Belt_1), 0{:}06) \circ \\ & \qquad Movement(Object, (7,0), Vel(Belt_1), 0{:}10) \circ StartTime(0{:}10)\,]\end{aligned} \tag{20}$$

(as sketched on the right hand side of Fig. 3).

Now, foundational axiom (2) in conjunction with (16) and (19) entails,

$$\begin{array}{l} \mathit{Trajectory}(\mathit{State}(\sigma), \mathit{Succ}(\mathit{Falls}(\mathit{Object}, 0{:}10), \mathit{State}(\sigma))) \vee \\ \mathit{Expect}(\mathit{Falls}(U, 0{:}06), \mathit{State}(\sigma)) \wedge \\ \quad \mathit{Trajectory}(\mathit{State}(\sigma), \mathit{Succ}(\mathit{Falls}(\mathit{Object}, 0{:}10), \\ \qquad\qquad\qquad\qquad \mathit{Succ}(\mathit{Falls}(U, 0{:}06), \mathit{State}(\sigma)))) \end{array}$$

This disjunction, along with (17) and (20), implies (11). ■

It is easy to see that no other plan than $\sigma = Do(Send(Container, 0{:}03), S_0)$ satisfies $Axioms_{Belts} \models (11)$. Hence, if a theorem prover which generates answer substitutions to query variables is applied to the deductive planning problem,

$$Axioms_{Belts} \models \exists s, p, t[Position(Container, p, s, t) \wedge Position(Object, p, s, t)]$$

then the unique answer (modulo equivalency) is

$$\{s/Do(Send(Container, 0{:}03), S_0), p/(t-3, 0)\}$$

5 Zeno's Paradox

In this section, we show how our theory allows to overcome yet another problem of the approach of [11], namely, Zeno's paradox. To allow the reader to appreciate the problem, let us recall the following example from [11]: Consider a sequence of natural actions $A(t)$ which are expected at all times $t = 1 - 2^{-n}$ for $n = 1, 2, \ldots$ If S_0 starts at time $t_0 = 0$, then the only legal situations are of the form $Do(A(\frac{2^n-1}{2^n}), \ldots, Do(A(\frac{1}{2}), S_0) \ldots)$. In other words, no action can occur after time $t = 1$.

Again, our uncoupling natural actions from the situations provides the key to a solution of this problem. Recall that in our theory a sequence of natural actions defines a trajectory. No restrictions have been imposed on the number of states in such a sequence. In particular, suppose a trajectory contains an infinite series of states in a finite time interval. An additional axiom may then state that also the limit of this series is on the trajectory. This enables conclusions about the evolution of the world state beyond the boundaries of the interval. Whenever a suitable (second-order) axiom on limits of infinite sequences of states can be provided, Zeno's paradox can thus be resolved.

To illustrate this use of limit axioms, let us formalize a small example domain. Suppose that $A(t)$ be the only natural action, that is,

$$Natural(a) \equiv \exists t.\, a = A(t) \tag{21}$$

This action shall have no effect apart from affecting the fluent $StartTime$:

$$\begin{array}{l} Expect(A(t), z) \supset \\ \quad \exists t_0.\, Succ(A(t), z) \circ StartTime(t_0) = z \circ StartTime(t) \end{array} \tag{22}$$

and A happens at all times $t = 1 - 2^{-n}$:

$$Expect(A(t), z) \equiv \exists n.\, t = 1 - \tfrac{1}{2^n} \wedge After(t, z) \tag{23}$$

where n is of sort natural number. The axiomatization thus far, including the three foundational axioms of Section 3, entails that all states $Z \circ StartTime(1 - 2^{-n})$ lie on the trajectory which roots in the initial state $State(S_0) = Z \circ StartTime(0)$. No conclusion can be drawn yet concerning

$ActualState(S_0, 1, z)$ and later states. Since our natural action has no further effects, it is a particularly simple second-order axiom on limits which suffices to overcome Zeno's paradox in this case:

$$\forall f : \mathbb{N} \mapsto \mathbb{R} \left[\begin{array}{l} Trajectory(z, z' \circ StartTime(f(0))) \wedge \\ \forall n\,[\, Trajectory(z, z' \circ StartTime(f(n))) \supset \\ \qquad Trajectory(z, z' \circ StartTime(f(n+1)))\,] \\ \supset Trajectory(z, z' \circ StartTime(\lim_{n \to \infty} f(n))) \end{array} \right] \tag{24}$$

That is, for each function f, if each state $z' \circ StartTime(f(n))$ (for $n \in \mathbb{N}$) is on a trajectory, then so is the state $z' \circ StartTime(\lim_{n \to \infty} f(n))$. This axiom allows to conclude that the state $Z \circ \lim_{n \to \infty}(1 - 2^{-n})$ lies on the trajectory rooted in $State(S_0)$. Since the sequence $\frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \ldots$ converges to 1, conclusions are thus enabled beyond the infinite natural actions.

Theorem 3 *Let $Axioms_{Zeno}$ be the foundational axioms of the Fluent Calculus with continuous change plus (21)–(24), then*

$$\begin{array}{rl} Axioms_{Zeno} \models & State(S_0) = z \circ StartTime(0) \supset \\ & \quad t \geq 1 \supset ActualState(S_0, t, z \circ StartTime(1)) \end{array}$$

Proof: Let Z be a *state* term such that $State(S_0) = Z \circ StartTime(0)$. From foundational axiom (2) it follows that

$$Trajectory(State(S_0), Z \circ StartTime(0)) \tag{25}$$

Furthermore, (22) and (23) imply that

$$\begin{array}{r} \forall n.\, Succ(A(1 - \frac{1}{2^{n+1}}), Z \circ StartTime(1 - \frac{1}{2^n})) = \\ Z \circ StartTime(1 - \frac{1}{2^{n+1}}) \end{array}$$

Hence, by (21), (23), and foundational axiom (2),

$$\begin{array}{l} \forall n\,[\, Trajectory(State(S_0), Z \circ StartTime(1 - \frac{1}{2^n})) \supset \\ \qquad Trajectory(State(S_0), Z \circ StartTime(1 - \frac{1}{2^{n+1}}))\,] \end{array}$$

From the latter along with (25) and limit axiom (24) we conclude that

$$Trajectory(State(S_0), Z \circ StartTime(1))$$

The claim follows from fact that $t \geq 1$ along with (21) and (23) implies

$$\neg \exists a\,[\, Natural(a) \wedge Expect(a, Z \circ StartTime(1)) \wedge Time(a) \leq t\,]$$

∎

More complicated infinite sequences of state updates, which have effects beyond the mere update of the fluent *StartTime*, require more sophisticated axioms on limits compared to (24). As soon as these can be provided, Zeno's paradox admits a solution within our formalism.

6 Discussion

On the basis of the predicate formalism of the Fluent Calculus we have developed a theory of reasoning about actions and planning in domains with natural actions and continuous change. By uncoupling natural actions from situations, we have overcome an unexpected limitation of an existing approach to modeling natural actions and continuous change with the Situation Calculus, [11, 4]. Incidentally, the crucial step towards a solution to the latter problem, the non-existence of plans in case of incomplete knowledge of world states, has also enabled solutions to Zeno's paradox. On the other hand, our formalism, as opposed to the approach of [11, 4], is not (yet) applicable to scenarios where (deliberative or natural) actions happen concurrently. Finding a good general theory of concurrency in the Fluent Calculus by which is preserved the solution to the inferential aspect of the Frame Problem, is still an important open research challenge.

An early attempt to combining logic-based action formalisms with continuous change is [12], where the frame problem for natural actions is solved by a non-monotonic approach: Continuity is the default, hence the occurrence of discontinuities (that is, changes in the set of differential equations characterizing a situation) is minimized. As shown in [10], this approach runs the risk of sanctioning anomalous models, which contain discontinuities that occur uncaused. This problem roots in the fact that simple minimization does not allows to tell apart discontinuities which have a cause. In later formalisms for reasoning about continuous change, [14, 15, 8, 9, 2], this problem has been solved by introducing the notion of natural actions (or: triggered events, etc.) as causes for discontinuities. In this way anomalous models are avoided by minimizing (usually by means of circumscription) the occurrence of causes rather than the discontinuities themselves. All of the formalisms just mentioned follow the principle of the Event Calculus [5] insofar as they model just a single linear time line—as opposed to [11, 4] and the approach taken in the present paper. The formal notion of linear time provides the basis both for suitably dealing with incomplete knowledge of the occurrence of natural actions and for solutions to Zeno's paradox by means of suitable limit axioms. On the other hand, a purely linear time structure requires abduction when it comes to solving planning problems [13] and so does not support the classical view of planning as a deductive task. A detailed comparison of abductive vs. deductive planning as well as arguments favoring the latter can be found in [11].

With the approach taken in the present paper we have introduced the novel concept of a tree of situations with trajectories, which combines branching (for deliberative actions) and linear (for natural actions) time. The former enables planning by deduction while the latter allows to accommodate both incomplete knowledge of natural action occurrences and infinitely many actions occurring in a finite interval.

References

[1] Cordell Green. Theorem proving by resolution as a basis for question-answering systems. *Machine Intelligence*, 4:183–205, 1969.

[2] Christoph S. Herrmann and Michael Thielscher. Reasoning about continuous processes. In B. Clancey and D. Weld, editors, *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 639–644, Portland, OR, August 1996. MIT Press.

[3] Steffen Hölldobler and Josef Schneeberger. A new deductive approach to planning. *New Generation Computing*, 8:225–244, 1990.

[4] Tod Kelley. Modeling complex systems in the situation calculus: A case study using the Dagstuhl steam boiler problem. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 26–37, Cambridge, MA, November 1996. Morgan Kaufmann.

[5] Robert Kowalski and M. Sergot. A logic based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[6] Hector J. Levesque. What is planning in the presence of sensing? In B. Clancey and D. Weld, editors, *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 1139–1146, Portland, OR, August 1996. MIT Press.

[7] John McCarthy. *Situations and Actions and Causal Laws.* Stanford Artificial Intelligence Project, Memo 2, 1963.

[8] Rob Miller. A case study in reasoning about actions and continuous change. In W. Wahlster, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, Budapest, Hungary, August 1996. John Wiley.

[9] Rob Miller and Murray Shanahan. Reasoning about discontinuities in the event calculus. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 63–74, Cambridge, MA, November 1996. Morgan Kaufmann.

[10] Manny Rayner. On the applicability of nonmonotonic logic to formal reasoning in continuous time. *Artificial Intelligence*, 49:345–360, 1991.

[11] Ray Reiter. Natural actions, concurrency and continuous time in the situation calculus. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 2–13, Cambridge, MA, November 1996. Morgan Kaufmann.

[12] Erik Sandewall. Combining logic and differential equations for describing real-world systems. In R. Brachman, H. J. Levesque, and R. Reiter, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 412–420, Toronto, Kanada, 1989. Morgan Kaufmann.

[13] Murray Shanahan. Prediction is deduction but explanation is abduction. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1055–1060, Detroit, MI, 1989.

[14] Murray Shanahan. Representing continuous change in the event calculus. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 598–603, 1990.

[15] Murray Shanahan. A circumscriptive calculus of events. *Artificial Intelligence*, 77:249–284, 1995.

[16] Michael Thielscher. Introduction to the Fluent Calculus. *Electronic Transactions on Artificial Intelligence*, 1998. URL: `http://www.ep.liu.se/ea/cis/1998/014/`.

[17] Michael Thielscher. Continuous processes in the fluent calculus. In *Hybrid Systems and AI: Modeling and Control of Discrete + Continuous Systems*, volume SS–99–05 of *AAAI Spring Symposia*, pages 186–191, Stanford University, March 1999. AAAI Press.

[18] Michael Thielscher. From Situation Calculus to Fluent Calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence*, 1999.