Continuous Processes in the Fluent Calculus

Michael Thielscher
Department of Computer Science
Dresden University of Technology
01062 Dresden (Germany)
mit@pikas.inf.tu-dresden.de

Abstract

Among the known predicate calculus formalisms for axiomatizing commonsense reasoning about actions, the Fluent Calculus stands out in offering a solution not only to the representational but also the inferential aspect of the fundamental Frame Problem. In this paper we extend this formalism to modeling hybrid systems, which involve both discrete and continuous change. We borrow basic notions from an existing extension of the Situation Calculus to this end, but depart from it in a crucial aspect: Exogenous events are uncoupled from the action sequence performed by the agent. In this way we solve the problem of non-existence of simple plans caused by incomplete knowledge of the ongoing processes, and we enable solutions to Zeno's paradox.

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 compute these non-effects. The Fluent Calculus, which roots in the logic programming formalism of (Hölldobler & Schneeberger 1990), provides an axiomatization strategy that particularly aims at the second aspect.

A new, expressive version of this calculus has recently been developed around the novel concept of so-called state update axioms (Thielscher 1998). In this paper we extend this formalism to explicitly reasoning about hybrid systems, which involve both discrete and continuous change. The first step towards this end is the insight that the fluents themselves, i.e., the atomic components of descriptions of world states, can be hybrid. The Fluent Calculus assumes a fluent to be stable in between the occurrence of two consecutive actions, and yet a fluent may internally represent an arbitrarily complex, continuous process (Herrmann & Thielscher 1996). A physical object x moving through the 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 continuous change, the 'process fluent' *Movement* itself remains unchanged unless some action causes a discontinuity. 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.

However, in a world full of ongoing processes, discontinuities may not only be caused by intervening agents but also be entailed by the laws of physics. Two moving objects, for instance, may eventually meet, which causes an 'autonomous' update of the world state. These exogenous events (or: *natural* actions (Reiter 1996)) can be specified just like deliberative actions by their preconditions and effects. Yet measures need to be taken to guarantee that it is always the action which is next in time, be it deliberative or natural, that leads to the correct state update (Sandewall 1989). In the context of the Situation Calculus, (Reiter 1996) proposes to combine into the situation argument both deliberative and natural actions, and to distinguish so-called legal situations $Do(a_n, Do(a_{n-1}, \ldots, Do(a_1, S_0) \ldots))$ where it is required for each situation $Do(a_i, \ldots, Do(a_1, S_0) \ldots)$ that no natural action is possible 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 will happen before.

Yet this merging natural and deliberative actions causes two problems: First, despite the existence of a simple sequence of deliberative actions by which an agent can solve a given planning problem, it may be impossible to state just such a plan due to incomplete knowledge of the ongoing processes. Suppose, for example, that a single deliberative action $A(t)$ at time $t = 2$ suffices to solve a planning problem with initial situation S_0 at time $t_0 = 0$. Suppose further that a non-interfering natural action $B(t)$ is known to happen at time $t = 1$ or at time $t = 3$. Then neither $Do(A(2), Do(B(1), S_0))$ nor $Do(A(2), S_0)$ is a provably legal situation. In other words, there does not exist a simple sequence of actions which can be proved to solve the planning problem.

Second, if infinitely many natural actions happen in finite time, then no conclusion can be drawn concerning later actions, be they deliberative or natural. (This is an instance of Zeno's paradox (Reiter 1996).) Consider, e.g., 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 $t = 1$.

It will be shown in the present paper how the first problem is solved by uncoupling natural actions from the action sequence performed by the agent. This improvement also allows to overcome the second problem in cases where suitable axioms on limits can be provided. Our exposition begins with a recapitulation of the Fluent Calculus as a solution to the inferential Frame Problem. Thereafter we extend the Fluent Calculus to reasoning about continuous change and natural actions. Our theory assumes deterministic worlds where actions or events cannot occur concurrently. We borrow some basic notions from (Reiter 1996) but uncouple natural events from situation terms. With the help of two examples we demonstrate how our theory can be used to overcome the two aforementioned problems. Finally, we illustrate our formalism with a scenario in which objects can be set into continuous motion by an agent and possibly interfere with other moving objects.

The Fluent Calculus

The purpose of the Fluent Calculus is to offer a solution to the inferential Frame Problem. The latter concerns each fluent value which, when proving a theorem, is needed in a situation other than the one for which it is given or in which it arises as an effect of an action or event. The Fluent Calculus addresses the inferential Frame Problem by specifying the effect of actions in terms of how an action modifies a state. A single so-called state update axiom (Thielscher 1998) always suffices to derive the entire change caused by the action in question.

The Fluent Calculus is a many-sorted second order language with equality, which includes sorts for fluents, actions, situations, and states. States are fluents connected via the binary function symbol "$\circ$", written in infix notation, which is assumed to be both associative and commutative, and to admit a unit element, denoted by $\emptyset$. A function $State(s)$ relates a situation s to the state of the world in that situation. As an example, suppose that about the initial state in some Blocks World scenario it is known that block A is on some block x, which in turn stands on the table, and that nothing is on top of block A or B. In the Fluent Calculus, this incomplete specification can be axiomatized as follows:

$$\begin{array}{l}\exists x, z\, [\, State(S_0) = On(A, x) \circ On(x, Table) \circ z \\ \quad \wedge\, \forall y, z'.\; z \neq On(y, A) \circ z' \wedge z \neq On(y, B) \circ z'\,]\end{array} \tag{1}$$

That is, of state $State(S_0)$ it is known that for some x both $On(A, x)$ and $On(x, Table)$ are true and that other facts z may hold—with the restriction that z does not include a fluent $On(y, A)$ nor $On(y, B)$.

Fundamental for any Fluent Calculus axiomatization is the axiom set $EUNA$ (the *extended unique names-assumption*) (Thielscher 1997). This set comprises the axioms AC1 (i.e., associativity, commutativity, and unit element) and axioms which entail inequality of two state terms whenever these are not AC1-unifiable, so that $EUNA$ is unification complete (Shepherdson 1992) wrt. terms of sort *state* and the equational theory AC1. In addition, we have the following foundational axiom,

$$\forall s, x, z.\; State(s) = x \circ x \circ z \supset x = \emptyset \tag{2}$$

by which double occurrences of fluents are prohibited in any state which is associated with a situation. (It will be explained shortly why "$\circ$" is not required to be idempotent to this end.) Finally, the Fluent Calculus uses the common *Holds* predicate, though not as part of the signature but as abbreviation of an equality sentence:

$$Holds(f, s) \stackrel{\text{def}}{=} \exists z.\; State(s) = f \circ z \tag{3}$$

So-called state update axioms specify the entire relation between the states at two consecutive situations. Deterministic actions with only direct and closed effects[1] give rise to the simplest form of state update axioms, where a mere equation relates a successor state $State(Do(A, s))$ to the preceding state $State(s)$. The general form of these state update axioms is,

$$\Delta(s) \supset State(Do(A, s)) \circ \vartheta^- = State(s) \circ \vartheta^+$$

where ϑ^- are the negative effects and ϑ^+ the positive effects, resp., of action A under condition $\Delta(s)$.[2] In this paper we will stick to the basic form of state update axioms, in order to focus on continuous processes. Beyond the simple case are disjunctive update axioms to model nondeterministic actions, and the axiomatization of indirect effects via causal propagation (Thielscher 1997). The perfect symmetry of the equation in the consequent of a state update axiom allows using it equally for reasoning forward and backward in time.

As an example, let the effect of an action called $Move(u, v, w)$ be that block u is moved away from the

[1] By closed effects we mean that an action does not have an unbounded number of direct effects.

[2] This scheme is the reason for not stipulating that "$\circ$" be idempotent, contrary to what one might intuitively expect. For if the function were idempotent, then the equation would not imply that $State(Do(A, s))$ does not include ϑ^-.

top of block v on top of block w. A suitable state update axiom is,

$$\begin{array}{l} Poss(Move(u,v,w),s) \supset \\ \quad State(Do(Move(u,v,w),s)) \circ On(u,v) = \\ \qquad\qquad State(s) \circ On(u,w) \end{array} \tag{4}$$

Let the precondition of *Move* be given by,

$$\begin{array}{l} Poss(Move(u,v,w),s) \equiv \\ \quad Holds(On(u,v),s) \wedge \\ \quad \neg\exists x.\, Holds(On(x,u),s) \wedge \neg\exists x.\, Holds(On(x,w),s) \end{array}$$

Consider, e.g., a scenario where the initial situation is described by formula (1), and suppose block A shall be moved away from x onto B. Then the expression $State(S_0)$ in the instance $\{u/A, v/x, w/B\}$ of state update axiom (4) can be replaced by the term which equals $State(S_0)$ according to (1). So doing yields, after evaluating $Poss(Move(A,x,B),S_0)$,

$$\begin{array}{l} \exists x,z.\, State(Do(Move(A,x,B),S_0)) \circ On(A,x) = \\ \qquad On(A,x) \circ On(x, Table) \circ z \circ On(A,B) \end{array}$$

This equational formula can be simplified to

$$\begin{array}{l} \exists x,z.\, State(Do(Move(A,x,B),S_0)) = \\ \qquad On(x, Table) \circ z \circ On(A,B) \end{array}$$

One thus obtains from an incomplete initial specification a still partial description of the successor state, which in particular includes the unaffected $On(x, Table)$. Hence, this fluent survived the computation of the effect of the action and so needs not be carried over by separate application of an axiom.

The inferential merits of the Fluent Calculus become apparent when contrasting state update axiom (4) to the successor state axiom which corresponds to our way of axiomatizing the Blocks World, viz.

$$\begin{array}{l} Poss(a,s) \supset \\ \quad On(u,w,Do(a,s)) \equiv \exists v.\, a = Move(u,v,w) \\ \qquad\qquad \vee\; On(u,w,s) \\ \qquad\qquad\quad \wedge \forall v.\, a \neq Move(u,w,v) \end{array} \tag{5}$$

To appreciate the computational difference between (4) and (5), consider the simple example of a planning problem (with incomplete information) specified by $On(A_1,B_1,S_0) \wedge \ldots \wedge On(A_{10},B_{10},S_0) \wedge \neg\exists x.\, On(x,A_1,S_0) \wedge \ldots \wedge \neg\exists x.\, On(x,A_{10},S_0) \wedge \neg\exists x.\, On(x,C_1,S_0) \wedge \ldots \wedge \neg\exists x.\, On(x,C_{10},S_0)$ along with the goal to reach a situation s which satisfies $On(A_1,C_1,s) \wedge \ldots \wedge On(A_{10},C_{10},s)$. A proof for a plan of minimal length to lead to the goal requires 145 instances of successor state axiom (5).[3] Formalized in the Fluent Calculus, a corresponding proof needs just 10 instances of state update axiom (4).

[3] If n is the number of blocks of each kind A, B, and C, then n^2 instances are needed to keep track of the locations of the blocks A_i, and $\frac{n}{2} \cdot (n-1)$ instances for the relevant information about the blocks C_i not (yet) being occupied.

Standard Interpretation of the Reals

For our extension of the Fluent Calculus to an explicit treatment of continuous change, we rely on the following standard interpretation, which we owe Cantor, of the reals, $\mathbb{R}$. A real number is a converging sequence of rational numbers $q_1, q_2, q_3, \ldots$ Converging means that $|q_{n+1} - q_n| \rightarrow 0$ for $n \rightarrow \infty$. The limit of such a converging sequence is denoted by $\lim_{n\rightarrow\infty} q_n$. Two real numbers $q_1, q_2, q_3, \ldots$ and $q'_1, q'_2, q'_3, \ldots$ are equal iff $|q_n - q'_n| \rightarrow 0$ for $n \rightarrow \infty$. Then $\lim_{n\rightarrow\infty} q_n = \lim_{n\rightarrow\infty} q'_n$. This standard interpretation of the reals in conjunction with suitable axioms on limits will, for instance, provide the theoretical foundation for overcoming Zeno's paradox.

Processes in the Fluent Calculus

The Fluent Calculus for hybrid systems includes the special sort *timepoint* with the real numbers as domain and with the standard interpretation as just described. This sort shall be accompanied by the standard arithmetic functions and comparison relations with their usual meaning.

In what follows, we will frequently talk about states that are not associated with a situation, which is why we introduce a convenient abbreviation similar to (3):

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

Each world state is required to include the special fluent $StartTime(t)$, denoting the starting time of the state. The start time must be unique.

$$\begin{array}{l} \exists t.\, HoldsIn(StartTime(t), State(s)) \\ \wedge\, [\, HoldsIn(StartTime(t_1), State(s)) \\ \quad \wedge\, HoldsIn(StartTime(t_2), State(s)) \supset t_1 = t_2 \,] \end{array}$$

As in (Reiter 1996), 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 $\alpha(\vec{x},t)$. A function $Time(a)$ maps actions to their execution time, that is, $Time(\alpha(\vec{x},t)) = t$.

The axiomatization of a natural action $\alpha(\vec{x},t)$ consists of two parts, its precondition and its effect. The former is specified by a definitional formula $Expect(\alpha(\vec{x},t), state) \equiv \Phi(\vec{x},t,state)$ where Φ defines the conditions under which $\alpha(\vec{x},t)$ is expected to automatically happen in state *state*, provided no other actions occur beforehand. The effect of a natural action is specified by $\Delta(\vec{x},t,state,state') \supset Event(state,state')$, where Δ defines the condition under which $\alpha(\vec{x},t)$ actually happens and causes an update from *state* to *state'*. In order to facilitate such specifications, we introduce the generic abbreviation,

$$\begin{array}{l} NearestEvent(a, state) \stackrel{\text{def}}{=} \\ \quad \forall a' \,[\, Natural(a') \wedge Expect(a', state) \\ \qquad \wedge\, Time(a') \leq Time(a) \supset a = a' \,] \end{array}$$

To this end, each domain specification is assumed to include an axiom $Natural(a) \equiv \Psi(a)$ where Ψ defines the conditions under which a is a relevant natural action.

A series of state updates by natural actions is called a trajectory. The predicate $Trajectory(state, state')$ is used to indicate that $state'$ eventually occurs on the trajectory starting with $state$:

$$\begin{aligned}&Trajectory(state, state)\\ \wedge\, [\, &Trajectory(state, state') \wedge Event(state', state'')\\ &\qquad \supset Trajectory(state, state'')\,]\end{aligned}$$

Notice that this foundational axiom allows more states on the trajectory apart from those that are reachable by a finite sequence of events. This feature is necessary to overcome Zeno's paradox; see the following but one section. Since we have confined ourselves to deterministic domains, no two different states with the same starting time shall be on a trajectory starting with the same state:

$$\begin{aligned}&Trajectory(state, state_1) \wedge Trajectory(state, state_2)\\ &\wedge HoldsIn(StartTime(t), state_1)\\ &\wedge HoldsIn(StartTime(t), state_2) \supset state_1 = state_2\end{aligned}$$

Decoupling natural actions from situation terms implies that the state $State(s)$, which is initially associated with a situation s, may evolve as time goes by. We therefore introduce the abbreviation $ActualState(s, t, state)$ to indicate that $state$ is the actual state in situation s at time t:

$$\begin{aligned}&ActualState(s, t, state) \stackrel{\text{def}}{=}\\ &\quad Trajectory(State(s), state) \wedge\\ &\quad \forall t_0\, [\, HoldsIn(StartTime(t_0), state) \supset t_0 \leq t\,] \wedge\\ &\quad \forall a\, [\, Natural(a) \wedge Expect(a, state) \supset Time(a) > t\,]\end{aligned}$$

Put in words, the actual state $state$ in situation s at time t lies on the trajectory of $State(s)$ and started at time $t_0 \leq t$, and future natural actions expected in $state$ will happen later.

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* predicate is furnished with an additional argument indicating the time at which a fluent is supposed to hold in a situation:

$$\begin{aligned}&Holds(f, s, t) \stackrel{\text{def}}{=}\\ &\quad \forall state\, [\, ActualState(s, t, state) \supset \exists z.\, state = f \circ z\,]\end{aligned}$$

It is furthermore convenient to use the abbreviations $After(t, s)$ and $After(t, state)$ to indicate that t denotes a timepoint later than the time at which situation s and state $state$, resp., came into being:

$$\begin{aligned}&After(t, s) \stackrel{\text{def}}{=}\\ &\quad \forall t_0\, [\, HoldsIn(StartTime(t_0), State(s)) \supset t > t_0\,]\\ &After(t, state) \stackrel{\text{def}}{=}\\ &\quad \forall t_0\, [\, HoldsIn(StartTime(t_0), state) \supset t > t_0\,]\end{aligned}$$

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

Existence of Simple Plans

In this section, we present a simple example which shows that in our formalization the problem does not arise of the non-existence of a plan due to incomplete knowledge of the ongoing processes. Let $A(t)$ be the only deliberative action and $B(t)$ the only natural action, that is,

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

Furthermore, let the only fluents be the nullary F and the unary $G(t)$, whose argument is of sort *timepoint*. Suppose that action $A(t)$ can only be performed at time $t = 2$ and if F is false at the time of execution. The effect shall be that F becomes true. Hence, this action is defined by the state update axiom,

$$\begin{aligned}&Poss(A(t), s) \supset\\ &\quad ActualState(s, t, state) \supset\\ &\quad\quad \exists t_0\, [\, State(Do(A(t), s)) \circ StartTime(t_0) =\\ &\quad\quad\quad state \circ F \circ StartTime(t)\,]\end{aligned}$$

along with the action precondition axiom,

$$Poss(A(t), s) \equiv t = 2 \wedge \neg Holds(F, s, t) \wedge After(t, s)$$

Natural action $B(t)$ happens whenever $G(t)$ holds:

$$\begin{aligned}&Expect(B(t), state) \equiv\\ &\quad HoldsIn(G(t), state) \wedge After(t, state)\end{aligned}$$

The action shall have no effects at all, that is,

$$\begin{aligned}&Expect(B(t), state) \wedge NearestEvent(B(t), state) \supset\\ &\quad \forall t_0\, [\, state = z \circ StartTime(t_0) \supset\\ &\quad\quad Event(state, z \circ StartTime(t))\,]\end{aligned}$$

Now, consider the planning problem with the goal to have F and where about the initial situation, starting at time 0, it is known that F is false and that $G(t)$ is true for either $t = 1$ or $t = 3$, that is,

$$\begin{aligned}&Holds(StartTime(0), S_0, 0)\\ \wedge\, &\neg Holds(F, S_0, 0)\\ \wedge\, &[\, Holds(G(t), S_0, 0) \supset t = 1 \vee t = 3\,]\\ \wedge\, &[\, Holds(G(1), S_0, 0) \equiv \neg Holds(G(3), S_0)\,]\end{aligned}$$

Obviously, a single performance of the deliberative action $A(2)$ suffices to solve the planning problem, no matter whether $B(t)$ occurs at time $t = 1$ or $t = 3$. Our formalism does indeed prove this simple plan a solution: Let Σ_{Simple} be the set of all axioms of this and the previous section plus foundational axiom (2) and *EUNA*.

Proposition 1 $\Sigma_{Simple} \models Holds(F, Do(A(2), S_0))$.

Zeno's Paradox

In this section, we illustrate how our formalism allows to overcome Zeno's paradox if suitable axioms on limits can be provided. Consider, to this end, a domain with the only natural action $A(t)$, that is,

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

This action shall have no effect, that is,

$$\begin{array}{l} \mathit{Expect}(A(t), \mathit{state}) \wedge \mathit{NearestEvent}(A(t), \mathit{state}) \supset \\ \quad \forall t_0 [\, \mathit{state} = z \circ \mathit{StartTime}(t_0) \supset \\ \qquad \mathit{Event}(\mathit{state}, z \circ \mathit{StartTime}(t)) \,] \end{array}$$

and it happens at all times $t = 1 - 2^{-n}$, that is,

$$\mathit{Expect}(A(t), \mathit{state}) \equiv \exists n.\, t = 1 - \frac{1}{2^n} \wedge \mathit{After}(t, \mathit{state})$$

where n is of sort natural number. The axiomatization so far, including the foundational axioms, entails that all states $z \circ \mathit{StartTime}(\frac{2^n-1}{2^n})$ lie on the trajectory which roots in $\mathit{State}(S_0) = z \circ \mathit{StartTime}(0)$. No conclusion can yet be drawn concerning $\mathit{ActualState}(S_0, 1, \mathit{state})$ nor later states.

Generally, Zeno's paradox is overcome by adding suitable axioms on limits which express that if an infinite sequence of states lies on a trajectory, then the latter also includes the limit of that sequence. Since the natural action in our particular example has no effects at all, the following simple second-order axiom suffices:

$$\forall f \left[\begin{array}{l} \mathit{Trajectory}(\mathit{state}, z \circ \mathit{StartTime}(f(0))) \\ \forall n.\, \mathit{Trajectory}(\mathit{state}, z \circ \mathit{StartTime}(f(n))) \\ \quad \supset \mathit{Trajectory}(\mathit{state}, z \circ \mathit{StartTime}(f(n+1))) \\ \supset \mathit{Trajectory}(\mathit{state}, z \circ \mathit{StartTime}(\lim_{n \to \infty} f(n))) \end{array} \right]$$

This axiom supports the additional conclusion that the state $z \circ \mathit{StartTime}(\lim_{n\to\infty}(\frac{2^n-1}{2^n}))$ lies on the trajectory rooted in $\mathit{State}(S_0) = z \circ \mathit{StartTime}(0)$. Since the sequence $\frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \ldots$ converges to 1, conclusions are thus enabled beyond the infinite series of natural actions. In particular, if Σ_{Zeno} is the set of all axioms of the present and previous but one section plus foundational axiom (2) and *EUNA*, then:

Proposition 2 *Σ_{Zeno} entails,*

$$\begin{array}{l} \mathit{State}(S_0) = z \circ \mathit{StartTime}(0) \supset \\ \quad t \geq 1 \supset \mathit{ActualState}(S_0, t, z \circ \mathit{StartTime}(1)) \end{array}$$

More complicated infinite sequences of state updates require more sophisticated axioms on limits. As soon as these can be provided, Zeno's paradox can be overcome.

A Complex Scenario

In this section, we present a more concrete scenario, where we model the detail of a production line depicted in Figure 1. Our goal is to find an axiomatization which allows to formalize the illustrated planning problem.

Let us first encode the background conditions of the scenario, that is the spatial velocity of the two conveyor belts and the location of the robot:

$$\begin{array}{c} \mathit{Vel}(\mathit{Belt}_1) = (1,0) \wedge \mathit{Vel}(\mathit{Belt}_2) = (0,-0.5) \\ \wedge\, \mathit{Loc}(\mathit{Robot}) = (2,1) \end{array}$$

The Fluents The only 'ordinary', i.e., stable fluent considered here is $\mathit{Has}(r,x)$, denoting whether agent r is in possession of object x. To this we add one kind of processes, namely, continuous movement in a two-dimensional space. This type of a process is represented by the fluent $\mathit{Movement}(x,p,v,t)$, an instance of which indicates that starting at time t, object x continuously moves away from location p with spatial velocity v. Being in a two-dimensional space, both p and v are binary vectors.

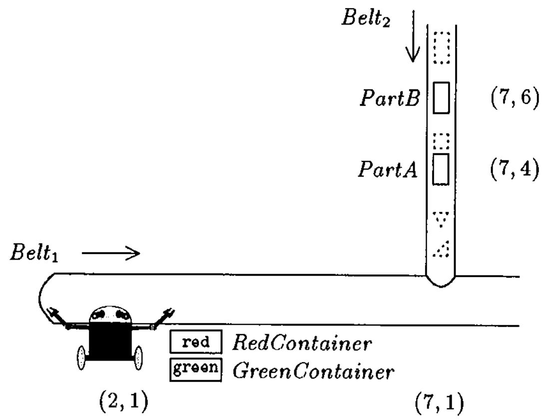

Figure 1: A robot located at coordinate position (2, 1) puts small containers on a conveyor belt, which are then transported to the right with a constant velocity of 1 unit per second. Incoming parts are moving on a second conveyor belt with a velocity of 0.5 units per second. Any incoming part falls onto Belt_1 as soon as it reaches the end of Belt_2 at coordinate position (7,1). If a container happens to be on Belt_1 at that time, then it takes the incoming part. Currently, there is nothing on Belt_1. Two incoming parts are *PartA* and *PartB*, currently at position (7,4) and (7,6), resp. The robot is given the planning problem to have *PartA* in the red container and *PartB* in the green one.

With the aid of the fluent and process types we can specify what is known to hold in the initial situation of our scenario:

$$\begin{array}{l} \exists z.\, \mathit{State}(S_0) = \mathit{Has}(\mathit{Robot}, \mathit{RedContainer}) \circ \\ \qquad \mathit{Has}(\mathit{Robot}, \mathit{GreenContainer}) \circ \\ \qquad \mathit{Movement}(\mathit{PartA}, (7,4), \mathit{Vel}(\mathit{Belt}_2), 0) \circ \\ \qquad \mathit{Movement}(\mathit{PartB}, (7,6), \mathit{Vel}(\mathit{Belt}_2), 0) \circ \\ \qquad \mathit{StartTime}(0) \circ z \end{array}$$

Put in words, our robot is in possession of the two containers, and the two incoming parts known of move continuously away from their respective starting point with the spatial velocity of Belt_2.

Other objects may be in motion in the initial situation, too, but only if they are on the incoming belt and move with the right velocity, that is,

$$\begin{array}{l} \mathit{Holds}(\mathit{Movement}(x,p,v,t), S_0, 0) \supset \\ \qquad \exists p_y.\, p = (7, p_y) \wedge v = \mathit{Vel}(\mathit{Belt}_2) \end{array}$$

Finally, we need the following state constraint, which ensures that at any time when a situation arises, any two existing movements must concern distinct objects located at different positions:

$$\begin{aligned}&Holds(Movement(x_1,p_1,v_1,t_1),s,t)\\ &\wedge Holds(Movement(x_2,p_2,v_2,t_2),s,t)\\ &\supset [x_1 = x_2 \equiv p_1 + v_1 \cdot (t-t_1) = p_2 + v_2 \cdot (t-t_2)]\end{aligned}$$

The Actions The only deliberative action we consider is putting objects on conveyor belts, represented by the action term $Put(r,x,b,t)$: At time t agent r puts object x on conveyor belt b. The preconditions for this action are that it is performed after the arising of the situation, the agent is in possession of the object in question, and no object happens to be on the belt at the same time and place. Formally,

$$\begin{aligned}&Poss(Put(r,x,b,t),s) \equiv\\ &\quad After(t,s) \wedge Holds(Has(r,x),s,t) \wedge\\ &\quad \forall [\, Holds(Movement(y,p,v,t'),s,t)\\ &\qquad \supset Loc(r) \neq p + v \cdot (t-t')\,]\end{aligned}$$

The effect of the action is that the agent loses possession of the object, which now moves along the direction and with the speed of the respective belt:

$$\begin{aligned}&Poss(Put(r,x,b,t),s) \supset ActualState(s,t,state) \supset\\ &\quad \exists t_0 .\, State(Do(Put(r,x,b,t),s)) \circ Has(r,x)\\ &\qquad\qquad \circ StartTime(t_0)\\ &= state \circ Movement(x, Loc(r), Vel(b), t) \circ StartTime(t)\end{aligned}$$

As a natural action we consider the taking of a part by a container, represented by the action term $Take(x,y,p,t)$: At time t object x takes object y at location p. This is the only natural action relevant to our scenario:

$$Natural(a) \equiv \exists [\, a = Take(x,y,p,t)\,]$$

This action can happen at any time t if the first object moves on $Belt_1$ and the second on $Belt_2$ and if the locations of the two objects coincide at time t. Formally,

$$\begin{aligned}&Expect(Take(x,y,p,t),state) \equiv\\ &\quad After(t,state) \wedge\\ &\quad HoldsIn(Movement(x,p_1,v_1,t_1),state) \wedge\\ &\quad HoldsIn(Movement(y,p_2,v_2,t_2),state) \wedge\\ &\quad \exists v_x .\, v_1 = (v_x,0) \wedge \exists v_y .\, v_2 = (0,v_y) \wedge\\ &\quad p = p_1 + v_1 \cdot (t-t_1) \wedge p = p_2 + v_2 \cdot (t-t_2)\end{aligned}$$

The effect of this natural action is that it terminates the two movements of container and object, resp., and that it initiates a new movement where the container has taken the part and the two together travel on $Belt_1$:

$$\begin{aligned}&Expect(Take(x,y,p,t),state)\\ &\wedge NearestEvent(Take(x,y,p,t),state) \supset\\ &\quad \forall t_0 [\, state = z \circ Movement(x,p_1,v_1,t_1) \circ\\ &\qquad Movement(y,p_2,v_2,t_2) \circ StartTime(t_0) \supset\\ &\quad Event(state, z \circ Movement(Inside(y,x),p,v_1,t) \circ\\ &\qquad StartTime(t))\,]\end{aligned}$$

Here, $Inside(x,y)$ represents the nested object which is produced by object x taking object y.

The Planning Problem Let Σ_{Robot} be the union of all axioms of this section plus the foundational axioms of the extended Fluent Calculus (including *EUNA*). We can then formalize our initial planning problem (c.f. Figure 1) as follows: Does there exist a situation S such that Σ_{Robot} entails,

$$\begin{aligned}&\exists p,v,t.\, Holds(Movement(Inside(PartA,\\ &\qquad RedContainer),p,v,t),S) \wedge\\ &\exists p,v,t.\, Holds(Movement(Inside(PartB,\\ &\qquad GreenContainer),p,v,t),S)\end{aligned} \tag{6}$$

The following result shows that the axiomatization indeed gives a solution to this problem, telling the robot at what times exactly it should put the two containers on the belt.

Theorem 3 *$\Sigma_{Robot} \models (6)$, where S is*

$$\begin{aligned}&Do(Put(Robot, GreenContainer, Belt_1, 5),\\ &\quad Do(Put(Robot, RedContainer, Belt_1, 1), S_0))\end{aligned}$$

Summary

We have extended an established predicate calculus formalism for reasoning about actions, namely, the Fluent Calculus, to modeling hybrid systems. We have borrowed basic notions from the Situation Calculus-based formalization proposed in (Reiter 1996) to this end, but we have also improved the latter by removing exogenous events from the situation term. In this way we have solved the problem of non-existence of simple plans caused by incomplete knowledge of the ongoing processes, and we have enabled solutions to Zeno's paradox. On the other hand, our formalism, as opposed to the approach of (Reiter 1996), is not yet applicable to scenarios where actions and events 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 problem.

References

Herrmann, C. S., and Thielscher, M. 1996. Reasoning about continuous processes. In *Proc. of AAAI*, 639–644.

Hölldobler, S., and Schneeberger, J. 1990. A new deductive approach to planning. *New Gener. Comput.* 8:225–244.

Reiter, R. 1996. Natural actions, concurrency and continuous time in the situation calculus. In *Proc. of KR*, 2–13.

Sandewall, E. 1989. Combining logic and differential equations for describing real-world systems. In *Proc. of KR*, 412–420.

Shepherdson, J. C. 1992. SLDNF-resolution with equality. *J. of Autom. Reason.* 8:297–306.

Thielscher, M. 1997. Ramification and causality. *Artif. Intell.* 89(1–2):317–364.

Thielscher, M. 1998. Towards state update axioms: Reifying successor state axioms. In *Proc. of JELIA*, vol. 1489 of *LNAI*, 248–263. (Also conditionally accepted for *Artif. Intell.*).