The Features-and-Fluents Semantics for the Fluent Calculus

Michael Thielscher and **Thomas Witkowski**
Department of Computer Science
Dresden University of Technology, Germany

Abstract

Based on an elaborate ontological taxonomy, the Features-and-Fluents framework provides an independent action semantics for assessing the range of applicability of action calculi. In this paper, we show how the fluent calculus can be used to capture the full range of phenomena in $\mathcal{K}$-**IA**, the broadest ontological class that has been fully formalized in (Sandewall 1994). To this end, we develop a significant extension of the fluent calculus for modeling actions with durations and with specific trajectories of changes. We present a provably correct translation of scenario descriptions from the Features-and-Fluents semantics into fluent calculus axiomatizations.

Introduction

Action formalisms are a core aspect of research in knowledge representation and reasoning. The classical approach, the situation calculus (McCarthy 1963), traces back to the beginning of Artificial Intelligence (McCarthy 1958) and has led to the fundamental frame problem (McCarthy & Hayes 1969). The concept of successor state axioms (Reiter 1991) has provided a first satisfactory solution to this problem in the situation calculus. By adding the explicit notion of a state, the situation calculus has been developed into the fluent calculus (Thielscher 2005b). Both solutions to the frame problem have been extended to capture a variety of phenomena, for example, nondeterministic actions (Lin 1996; Thielscher 2000a). The two calculi employ pure classical logic and thus are amenable to its standard semantics. However, an analysis of their range of applicability based on an independent, equally expressive action semantics has not been carried out.

The Features-and-Fluents framework of (Sandewall 1994) constitutes such an independent action semantics, which includes an elaborate ontological taxonomy comprising a variety of aspects like conditional effects, nondeterministic outcomes, and actions with durations. This semantics has been used to assess the range of applicability of nonmonotonic solutions to the frame problem based on preferential entailment. Yet it has been an open problem to show that a standard monotonic approach, like the situation calculus or the fluent calculus, is expressive enough to be applicable to the full test suite of example reasoning problems in (Sandewall 1994), let alone to be provably sound and complete wrt. one of the more expressive ontological classes in the taxonomy.

In this paper, we present a version of the fluent calculus that is sufficiently expressive to capture $\mathcal{K}$-**IA**, which is the broadest class that has been rigorously formalized and intensively studied in (Sandewall 1994) and in which correct knowledge and a fully inertial world is assumed. To this end, we develop a significant extension of the basic fluent calculus by introducing an explicit model for the duration of actions and for trajectories of changes. On this basis, we present a translation function that maps any $\mathcal{K}$-**IA** scenario into a fluent calculus axiomatization, and we prove that the intended models of the former coincide with the classical models of the latter. In this way the fluent calculus provides a sound and complete reasoning method for this ontological class of the Features-and-Fluents semantics. As a simple consequence of this result, the fluent calculus is shown to handle the entire test suite of example problems in (Sandewall 1994).

The rest of the paper is organized as follows: We begin by giving a brief introduction to both the Features-and-Fluents semantics and the fluent calculus. Thereafter, we develop an extension of the fluent calculus for representing and reasoning about the trajectories of changes for actions with explicit durations. We then present a mapping from $\mathcal{K}$-**IA** scenarios into the extended fluent calculus and prove its soundness and completeness. A summary and discussion concludes the paper.

The Ego-World-Semantics

In the following, we give a condensed introduction to the Features-and-Fluents framework; we refer to (Sandewall 1994) for more details.

Inhabited dynamical systems

An *inhabited dynamical system* (IDS) is a collection of possibly related objects whose state changes over time influenced by an *ego*. At each point in time, an IDS is in a *state*, which is an assignment of values to a given set of *features*. The domain of features will be represented by the symbol $\mathcal{F}$, the domain of states by $\mathcal{R}$, and the domain of timepoints by $\mathcal{T}$, which throughout the paper will be the set of nonnegative integers. An *IDS history* is a function $R(t, f)$ which, for a given timepoint t and a feature f, assigns an appropriate value to that feature.

In the IDS reality, an *event occurrence* happens over an interval of time and is denoted by $\langle s, E, t\rangle \in \mathcal{T} \times \mathcal{E} \times \mathcal{T}$, where $s < t$ and $\mathcal{E}$ denotes the domain of occurrence designators (also called *actions*). A *development* of an IDS can be understood as a tuple which contains all information about occurrences, feature-values, and timepoints during the run of the system.

Definition 1. *A 5-tuple $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C}\rangle$ is a* development *of an IDS if*

- $\mathcal{B} \subseteq \mathcal{T}$ *is a finite set of timepoints, whose members are called* breakpoints*; the largest member of $\mathcal{B}$ is called* now *and is written $n_{\mathcal{B}}$;*
- M *is a* valuation *which maps every object constant to an element of a given* object domain $\mathcal{O}$ *(see also Definition 2 below) and every timepoint constant to a nonnegative integer;*
- R *is an IDS history defined up to timepoint $n_{\mathcal{B}}$;*
- $\mathcal{A}$*, the* past action set*, is a set of occurrences $\langle s, E, t\rangle$, where $s, t \in \mathcal{B}$;*
- $\mathcal{C}$*, the* current action set*, is a set of pairs $\langle s, E\rangle$, where $s \in \mathcal{B}$.*

The interaction between an ego and a world is understood as a game, where both the ego and the world take turns and where, starting from an initial state r_0 of the world at time zero, they construct a development. For a given valuation M, the initial development is defined as $\langle \{0\}, M, \{0 \mapsto r_0\}, \emptyset, \emptyset\rangle$. The ego, when it makes a move, either starts an action by adding $\langle s, E\rangle$ to $\mathcal{C}$ (with $n_{\mathcal{B}} < s$) and s to $\mathcal{B}$, or the ego terminates an action by removing $\langle s, E\rangle$ from $\mathcal{C}$ (with $s < n_{\mathcal{B}}$) and a corresponding addition of $\langle s, E, n_{\mathcal{B}}\rangle$ to $\mathcal{A}$. Afterwards, the world extrapolates what happens as a result of the current state of the world and the actions that the ego has initiated. It can either add exactly one further member n to $\mathcal{B}$ and construct a new development as a revision up to n, or it can leave the current state unchanged and extend the history R so that it becomes complete.

Ontological taxonomy

The definition of an ontological taxonomy over the set of worlds and scenario descriptions allows one to classify worlds and problems which fulfill certain epistemological assumptions. While the Features-and-Fluents framework is a very general semantics, the broadest class that has been fully formalized and intensively studied in (Sandewall 1994) is denoted by $\mathcal{K}$-**IA**, where $\mathcal{K}$ means that the ego has correct knowledge of the world, I means that the world is inertial, and A stands for alternative effects of actions (e.g., conditional or nondeterministic). With this general class on top of the hierarchy, a taxonomy is built by restrictions to sub-specialties, like complete knowledge of the initial state, single-step actions, equidurational change, deterministic actions, etc.

`scd1`	$[s_1, t_1]$*Load*
`scd2`	$[s_2, t_2]$*Spin*
`scd3`	$[s_3, t_3]$*Fire*
`scd4`	$t_1 < s_2 \wedge t_2 < s_3$
`obs1`	$[0]\mathit{alive}\hat{=}\mathtt{T} \wedge [t_3]\mathit{alive}\hat{=}\mathtt{F}$

Figure 1: A chronicle for a variant of the Russian Turkey example.

Chronicles

In the following, we formally define scenario descriptions, called *chronicles*, in $\mathcal{K}$-**IA**. This requires some preparatory definitions.

A *feature expression* is of the form $F(\omega_1, \ldots, \omega_n)$, where $F^n \in \mathcal{F}$ is an n-ary feature symbol and each ω_i is either an object constant or an object variable ($n \geq 0$). Using the standard logical connectives, *fluent formulas* are built from atomic formulas of the form $f\hat{=}\mathcal{X}$, meaning that feature expression f has one of the values $\mathcal{X} = \{x_1, \ldots, x_n\}$ ($n \geq 1$). If $\mathcal{X} = \{x\}$ is a singleton, then we simply write $f\hat{=}x$. Using the standard logical connectives, *logic formulas* are built from *elementary formulas*, which are

- relational formulas among object constants or timepoints, which are either timepoint constants or members of $\mathcal{T}$ (i.e., nonnegative integers); or
- expressions of the form $[\tau]\phi$, where τ is a timepoint expression (i.e., an arithmetic term over timepoints) and ϕ is a fluent formula.

An example of a logic formula is `obs1` in Figure 1, where *alive* is a nullary feature symbol with domain $\{\mathtt{F}, \mathtt{T}\}$. An *action statement* $[\varsigma, \tau]\varepsilon$ consists of variable-free timepoint expressions ς and τ and a variable-free occurrence expression $\varepsilon = A(\omega_1, \ldots, \omega_n)$, where A is an n-ary action symbol and each ω_i is an object constant. An example of an action statement is `scd1` in Figure 1.

Definition 2. *A 4-tuple $\langle \mathcal{O}, \langle \mathtt{Infl}, \mathtt{Trajs}\rangle, \mathrm{SCD}, \mathrm{OBS}\rangle$ is a* chronicle *if*

- $\mathcal{O}$ *is an object domain;*
- *the* world description $\langle \mathtt{Infl}, \mathtt{Trajs}\rangle$ *consists of*
  - *a mapping $\mathtt{Infl}(E, r)$ from actions $E \in \mathcal{E}$ and partial states*[1] *r to finite subsets of $\mathcal{F}$ (defining the set of features that are potentially affected by E if executed in a state of the world that satisfies r), and*
  - *a finite, non-empty set $\mathtt{Trajs}(E, r)$ whose elements are finite and non-empty sequences of partial states (the* trajectories*) assigning values to the features in $\mathtt{Infl}(E, r)$;*
- *the* schedule set SCD *consists of action statements along with* timing statements*, which are formulas that use only timepoint expressions;*
- *the* observation set OBS *is a set of logic formulas.*

[1] A partial state is an assignment of values to a finite subset of all features occurring in a chronicle.

Partial starting state r	$\texttt{Infl}(\mathit{Load}, r)$	$\texttt{Trajs}(\mathit{Load}, r)$
$\{\mathit{shot} : \texttt{0}\}$	$\{\mathit{shot}\}$	$\langle\{\mathit{shot} : \texttt{2}\}\rangle$
$\{\mathit{shot} : \texttt{1}\}$	$\{\mathit{shot}\}$	$\langle\{\mathit{shot} : \texttt{2}\}\rangle$
$\{\mathit{shot} : \texttt{2}\}$	$\emptyset$	$\langle\emptyset\rangle$

Partial starting state r	$\texttt{Infl}(\mathit{Spin}, r)$	$\texttt{Trajs}(\mathit{Spin}, r)$
$\{\mathit{shot} : \texttt{0}\}$	$\emptyset$	$\langle\emptyset\rangle$
$\{\mathit{shot} : \texttt{1}\}$	$\{\mathit{shot}\}$	$\langle\{\mathit{shot} : \texttt{0}\}\rangle, \langle\{\mathit{shot} : \texttt{1}\}\rangle$
$\{\mathit{shot} : \texttt{2}\}$	$\{\mathit{shot}\}$	$\langle\{\mathit{shot} : \texttt{0}\}\rangle, \langle\{\mathit{shot} : \texttt{2}\}\rangle$

Partial starting state r	$\texttt{Infl}(\mathit{Fire}, r)$	$\texttt{Trajs}(\mathit{Fire}, r)$
$\{\mathit{alive} : \texttt{T}, \mathit{shot} : \texttt{0}\}$	$\emptyset$	$\langle\emptyset\rangle$
$\{\mathit{alive} : \texttt{T}, \mathit{shot} : \texttt{1}\}$	$\{\mathit{alive}, \mathit{shot}\}$	$\langle\{\mathit{alive} : \texttt{T}, \mathit{shot} : \texttt{0}\},$ $\{\mathit{alive} : \texttt{F}, \mathit{shot} : \texttt{0}\}\rangle$
$\{\mathit{alive} : \texttt{T}, \mathit{shot} : \texttt{2}\}$	$\{\mathit{alive}, \mathit{shot}\}$	$\langle\{\mathit{alive} : \texttt{T}, \mathit{shot} : \texttt{1}\},$ $\{\mathit{alive} : \texttt{F}, \mathit{shot} : \texttt{1}\}\rangle$
$\{\mathit{alive} : \texttt{F}, \mathit{shot} : \texttt{0}\}$	$\emptyset$	$\langle\emptyset\rangle$
$\{\mathit{alive} : \texttt{F}, \mathit{shot} : \texttt{1}\}$	$\{\mathit{shot}\}$	$\langle\{\mathit{shot} : \texttt{0}\}\rangle$
$\{\mathit{alive} : \texttt{F}, \mathit{shot} : \texttt{2}\}$	$\{\mathit{shot}\}$	$\langle\{\mathit{shot} : \texttt{1}\}\rangle$

Figure 2: Trajectories for our variant of the Russian Turkey domain: Loading the gun always has the effect of providing for two shots; spinning has the nondeterministic effect that either the gun becomes unloaded, or the number of shots remains unchanged; and firing a loaded gun has the consecutive effect that, first, the number of shots is decreased and, then, that the turkey dies provided it was alive.

Example 1. *Consider a vocabulary for a variant of the Russian Turkey scenario—an extension of the well-known Yale Shooting problem (Hanks & McDermott 1987)—comprising the (nullary) features* $\mathit{alive}^0 : \{\texttt{T}, \texttt{F}\}$ *and* $\mathit{shot}^0 : \{\texttt{0}, \texttt{1}, \texttt{2}\}$, *where the latter describes the number of possible shots without reloading the gun. Let the occurrence vocabulary consist of the nullary symbols Load, Spin, and Fire. The laws in Figure 1 along with the empty object domain* $\mathcal{O} = \emptyset$ *and the trajectories in Figure 2 constitute a chronicle.*

The specification of multiple trajectories for an action and a partial state can have two reasons: The corresponding action may be nondeterministic, like *Spin*, or the (order of the) change of the fluent's values is indeterminate.

On top of the trajectory description language, (Sandewall 1994) includes an informal definition of a language for effect laws and their translation into a trajectory table. A variant of this language, along with a more formal definition of its semantics, has been presented in (Sandewall 1998). We refrain from giving a more thorough treatment of this issue, because our translation into the fluent calculus will take a trajectory description directly as input.

Semantics

For a given action E and a partial state r, let r' be the restriction of r to the features in $\texttt{Infl}(E, r)$, and let $\langle r'_1, \ldots, r'_k \rangle$ be a member of $\texttt{Trajs}(E, r)$. This trajectory represents an execution of the form $\langle s, E, s + k \rangle$, where the IDS system at times $s + 1, \ldots, s + k$ is in states $r_1, \ldots, r_k$ such that each r_i is obtained from r'_i by choosing, for all features not in $\texttt{Infl}(E, r)$, the value they have in the full state in which the execution of E begins. In this way, the trajectory $\langle r'_1, \ldots, r'_k \rangle$ specifies the successive change in the members of $\texttt{Infl}(E, r)$ while all other features remain unchanged. Different initial states may lead to different execution sequences, but also the same initial state may lead to different execution sequences if there is more than one trajectory.

Models in the Features-and-Fluents framework are developments resulting from games between an ego and a world for a given correct valuation. A valuation (cf. Definition 1) is *correct* with respect to a given $\mathcal{K}$-**IA** chronicle $\Upsilon = \langle \mathcal{O}, \langle \texttt{Infl}, \texttt{Trajs} \rangle, \text{SCD}, \text{OBS} \rangle$ if it satisfies the following conditions:

- all object constants occurring in Υ are mapped to an element from $\mathcal{O}$;
- all timepoint constants occurring in Υ are mapped to a nonnegative integer;
- for every $[s, t]E$ in SCD, $M[s] < M[t]$ holds;
- for every distinct pair $[s_i, t_i]E_i$ and $[s_j, t_j]E_j$ in SCD, either $M[t_i] \leq M[s_j]$ or $M[t_j] \leq M[s_i]$ holds;
- M satisfies all inequations in SCD.

In the following, if α is an arbitrary formula and M a valuation, we write $M[\alpha]$ for the modified formula obtained by replacing each constant symbol in α by its value according to M.

Definition 3. *Let* $\Upsilon = \langle \mathcal{O}, \langle \texttt{Infl}, \texttt{Trajs} \rangle, \text{SCD}, \text{OBS} \rangle$ *be a* $\mathcal{K}$-**IA** *chronicle. A development* $\langle \mathcal{B}, M, R, \mathcal{A}, \mathcal{C} \rangle$ *is an* intended model *for* Υ *if*

t	$R(alive, t)$	$R(shot, t)$	Action
0	T	$\{0, 1, 2\}$	
1	T	$\{0, 1, 2\}$	*Load*
2	T	2	
3	T	2	
4	T	2	*Spin*
5	T	2	
6	T	2	*Fire*
7	T	1	
8	F	1	

Figure 3: Three IDS histories for the $\mathcal{K}$-**IA** chronicle from Example 1: The only difference is the state at timepoints 0 and 1, depending on the initial number of shots.

- *the valuation M is correct with respect to Υ;*
- $\mathcal{A} = M[\text{SCD}]$, *that is, the actions that occurred during the game are exactly the actions that are mentioned in the schedule;*
- *whenever the ego starts an action by adding $\langle n, E\rangle$ to $\mathcal{C}'$ in an intermediate development $\langle \mathcal{B}', M', R', \mathcal{A}', \mathcal{C}'\rangle$, the world chooses both a partial state r such that $R'(n)$ satisfies r and a member of $\mathtt{Trajs}(E, r)$ of the form $\langle r'_1, \ldots, r'_k\rangle$ to update the history R' up to timepoint $m = n + k$;*
- *the history R satisfies the observations in* OBS.

The set of all intended models of a $\mathcal{K}$-**IA** chronicle is denoted by $\Sigma_{\mathbf{IA}}(\Upsilon)$. In the following, we will consider intended models to be characterized just by the pair $\langle M, R\rangle$ of the corresponding development.

Example 1. *Consider the Russian Turkey chronicle from above. A possible correct valuation M for this chronicle is given by $M[s_1] = 1,\ M[t_1] = 2,\ M[s_2] = 4,\ M[t_2] = 5,\ M[s_3] = 6,\ M[t_3] = 8$.*

Under this valuation there are three intended models: To begin with, all intended models satisfy $R(alive, 0) = \mathtt{T}$ due to obs1. *On the other hand, the chronicle does not entail any restriction for the value of the second feature, so that there are three initial states with $shot \widehat{=} 0$, $shot \widehat{=} 1$, and $shot \widehat{=} 2$, respectively. Loading the gun at timepoint 1 necessarily results in $shot \widehat{=} 2$ at timepoint 2. The given valuation says that Spin is executed at timepoint 4. This being a nondeterministic actions, there are both developments where the gun becomes unloaded and developments where this action has no effects. But only the latter can be completed to intended models, because with an unloaded gun the subsequent Fire action would not have the effect to kill the turkey, contrary to what* obs1 *says. The trajectory for action Fire then requires that at timepoint 7 the value of shot is decreased and at timepoint 8 the turkey is no longer alive. The complete histories of the three intended models under the given valuation are summarized in Figure 3. Independent of the concrete valuation, all intended models for the Russian Turkey chronicle entail $[t_2] shot \widehat{=} 2$, that is, the gun did not become unloaded after spinning.*

The Fluent Calculus with Trajectories

We begin by recalling the basic notions and notations of the fluent calculus.

Basic fluent calculus

The fluent calculus is a many-sorted predicate logic language which includes the two standard sorts FLUENT and STATE. A *fluent* is an atomic property which can change its value over time. Every fluent is also a (singleton) STATE. The empty state is denoted by the constant $\emptyset$: STATE, and states are built using the binary function $\circ$: STATE $\times$ STATE $\mapsto$ STATE. The fundamental macro $Holds(f, z)$ denotes that a fluent f holds in a state z:

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

The *foundational axioms* of the fluent calculus are:[2]

- Associativity and commutativity,
$$(z_1 \circ z_2) \circ z_3 = z_1 \circ (z_2 \circ z_3)$$
$$z_1 \circ z_2 = z_2 \circ z_1$$
- Empty state axiom and irreducibility,
$$\neg Holds(f, \emptyset)$$
$$Holds(f_1, f) \rightarrow f_1 = f$$
- Decomposition and state equivalence,
$$Holds(f, z_1 \circ z_2) \rightarrow Holds(f, z_1) \vee Holds(f, z_2)$$
$$(\forall f)(Holds(f, z_1) \leftrightarrow Holds(f, z_2)) \rightarrow z_1 = z_2$$
- Existence of states,
$$(\forall P)(\exists z)(\forall f)(Holds(f, z) \leftrightarrow P(f)) \qquad (1)$$
where P is a second-order predicate variable of sort FLUENT.

These axioms essentially characterize states as (non-nested) sets of fluents. Second-order axiom (1) guarantees the existence of a state for any combination of fluents.

Effects of actions are axiomatized in the fluent calculus on the basis of two macros for, respectively, removing and adding fluents to states:

$$z_1 - f = z_2 \stackrel{\text{def}}{=} (z_2 = z_1 \vee z_2 \circ f = z_1) \wedge \neg Holds(f, z_2)$$
$$z_1 + f = z_2 \stackrel{\text{def}}{=} z_2 = z_1 \circ f$$

These macros are straightforwardly generalized to removal and addition of finitely many fluents. The following basic result of the fluent calculus (see, e.g., (Thielscher 2005b)) lays the foundation for a solution to the frame problem in classical logic:

Proposition 1. *Let $effects^- = f_1 \circ \ldots \circ f_m$ and $effects^+ = f_{m+1} \circ \ldots \circ f_n$ be two finite state terms ($0 \leq m \leq n$). Under the foundational axioms of the fluent calculus, equation $z_2 = z_1 - effects^- + effects^+$ implies*

$$Holds(f, z_2) \leftrightarrow Holds(f, effects^+) \vee [Holds(f, z_1) \wedge \neg Holds(f, effects^-)]$$

[2] Variables of sort FLUENT and STATE will be denoted, respectively, by the letter f and z, possibly with sub- or superscript.

Inherited from the situation calculus, *situations* (i.e., terms of sort SIT) in the fluent calculus are sequence of terms of sort ACTION, where the initial situation is usually denoted by S_0 and the function *Do*: ACTION×SIT $\mapsto$ SIT is used to compose actions and situations. The predicate *Poss*: ACTION × STATE is used to define preconditions of actions.

Actions with duration

The basic fluent calculus does not contain an explicit model of a time structure. Since the Features-and-Fluents semantics makes use of a discrete time domain, we extend the fluent calculus by the sort TIMEPOINT, which will be interpreted as the nonnegative integers in this paper. We will use the arithmetic functions $+$, $-$, $*$ and the relations $<$, $\leq$ and rely on their standard interpretation. Variables of the sort TIMEPOINT are denoted by the letter t, possibly with sub- or superscript.

In order to model actions with durations, the standard function *Do* is extended to *Do*: ACTION×TIMEPOINT× TIMEPOINT × SIT $\mapsto$ SIT. The intended meaning is that $Do(a, t_1, t_2, s)$ denotes the situation after executing the action a in situation s, with starting timepoint t_1 and termination timepoint t_2.

For every situation s, the macro $start(s)$ denotes the time of termination of the last action in situation s:

$$\begin{aligned} start(s) = t \stackrel{\text{def}}{=} \; & s = S_0 \wedge t = 0 \\ & \vee (\exists a, s', t')\, s = Do(a, t', t, s') \end{aligned}$$

For the sake of simplicity, we will slightly abuse notation and use $start(s)$ directly as term inside of formulas Φ, with the intended meaning $(\exists t)\,(start(s) = t \wedge \Phi')$, where Φ' is as Φ but with $start(s)$ replaced by t.

The preconditions for executing an action require that the action does not occur prior to the beginning of a situation and that the action can only terminate after it has started. In order to axiomatize preconditions for actions with explicit duration, the standard predicate *Poss* is extended so as to include both the starting time of the action and that of its termination.

Definition 4. *A* precondition axiom *for an action* $A(\vec{x})$ *is a formula of the form*

$$\begin{aligned} & Poss(A(\vec{x}), t_1, t_2, s) \leftrightarrow \\ & \quad (start(s) \leq t_1 < t_2) \wedge \Phi(\vec{x}, t_1, t_2, s) \end{aligned}$$

where t_1 *and* t_2 *are timepoint expressions and* Φ *is a formula with free variables* $\vec{x}, t_1, t_2, s$.

Trajectories

In order to capture the full expressivity of the Features-and-Fluents semantics, a further extension is required for representing and reasoning about trajectories. For this purpose, we introduce the function *TState* : SIT × TIMEPOINT $\mapsto$ STATE. The expression $TState(s, t)$ denotes the state in situation s at time t. In every situation, the state is assumed to remain unchanged after the last action, and consecutive situations are assumed to have the same history:

$$(\forall t) \;\; TState(S_0, t) = TState(S_0, 0) \quad (2)$$

$$(\forall t > t_2) \;\; TState(Do(a, t_1, t_2, s), t) = TState(Do(a, t_1, t_2, s), t_2) \quad (3)$$

$$(\forall t \leq t_1) \;\; TState(Do(a, t_1, t_2, s), t) = TState(s, t) \quad (4)$$

An extended *Holds*-macro allows us to state that a fluent holds at a given timepoint in a given situation:

$$Holds(f, t, s) \stackrel{\text{def}}{=} Holds(f, TState(s, t))$$

The effects of actions are specified in the extended fluent calculus by a generalized form of *state update axioms*, which define how and when the affected fluents change.

Definition 5. *A* state update axiom *for an action* $A(\vec{x})$ *is a formula of the form*

$$\begin{aligned} & Poss(A(\vec{x}), t_1, t_2, s) \rightarrow \\ & \quad \Delta_1(s) \wedge \Gamma_1[TState(Do(A(\vec{x}), t_1, t_2, s), t)] \\ & \quad \vee \ldots \vee \\ & \quad \Delta_n(s) \wedge \Gamma_n[TState(Do(A(\vec{x}), t_1, t_2, s), t)] \end{aligned}$$

where each sub-formula Γ_i *describes a possible trajectory under condition* Δ_i *(*$1 \leq i \leq n$*).*

Example 2. *Consider the fluents Alive and Shot* : $\mathbb{N}$ *along with the two domain constraints*

$$\begin{aligned} & (\exists\, 0 \leq n \leq 2)\; Holds(Shot(n), t, s) \\ & Holds(Shot(n_1), t, s) \wedge Holds(Shot(n_2), t, s) \rightarrow \\ & \qquad n_1 = n_2 \end{aligned}$$

Our variant of the Russian Turkey domain of Example 1 may then be axiomatized in the extended fluent calculus by the following precondition axioms:

$$\begin{aligned} & Poss(Load, t_1, t_2, s) \leftrightarrow \\ & \quad (start(s) \leq t_1 < t_2) \wedge t_2 = t_1 + 1 \\ & Poss(Spin, t_1, t_2, s) \leftrightarrow \\ & \quad (start(s) \leq t_1 < t_2) \wedge t_2 = t_1 + 1 \\ & Poss(Fire, t_1, t_2, s) \leftrightarrow \\ & \quad (start(s) \leq t_1 < t_2) \\ & \qquad \wedge (Holds(Alive, t_1, s) \wedge \neg Holds(Shot(0), t_1, s) \\ & \qquad\quad \rightarrow t_2 = t_1 + 2) \\ & \qquad \wedge (\neg Holds(Alive, t_1, s) \vee Holds(Shot(0), t_1, s) \\ & \qquad\quad \rightarrow t_2 = t_1 + 1) \end{aligned}$$

Put in words, loading and spinning the gun always takes one time unit while firing takes either two or one time units, depending on whether the turkey is alive and the gun is loaded.

The state update axioms are as follows:

$$
\begin{array}{l}
Poss(Load, t_1, t_2, s) \rightarrow \\
\quad (\exists n)\,(Holds(Shot(n), t_1, s) \wedge \\
\qquad\quad TState(Do(Load, t_1, t_2, s), t_1+1) = \\
\qquad\qquad TState(s, t_1) - Shot(n) + Shot(2))
\end{array}
$$

$$
\begin{array}{l}
Poss(Spin, t_1, t_2, s) \rightarrow \\
\quad (\exists n)\,(Holds(Shot(n), t_1, s) \wedge n > 0 \wedge \\
\qquad\quad TState(Do(Spin, t_1, t_2, s), t_1+1) = \\
\qquad\qquad TState(s, t_1) - Shot(n) + Shot(0)) \\
\quad \vee \\
\quad TState(Do(Spin, t_1, t_2, s), t_1+1) = TState(s, t_1)
\end{array}
$$

$$
\begin{array}{l}
Poss(Fire, t_1, t_2, s) \rightarrow \\
\quad (\exists n)(Holds(Alive, t_1, s) \wedge \\
\qquad\quad Holds(Shot(n), t_1, s) \wedge n > 0 \wedge \\
\qquad\quad TState(Do(Fire, t_1, t_2, s), t_1+1) = \\
\qquad\qquad TState(s, t_1) - Shot(n) + Shot(n-1) \\
\qquad\quad \wedge \\
\qquad\quad TState(Do(Fire, t_1, t_2, s), t_1+2) = \\
\qquad\qquad TState(Do(Fire, t_1, t_2, s), t_1+1) - Alive) \\
\quad \vee \\
\quad (\exists n)(\neg Holds(Alive, t_1, s) \wedge \\
\qquad\quad Holds(Shot(n), t_1, s) \wedge n > 0 \wedge \\
\qquad\quad TState(Do(Fire, t_1, t_2, s), t_1+1) = \\
\qquad\qquad TState(s, t_1) - Shot(n) + Shot(n-1)) \\
\quad \vee \\
\quad Holds(Shot(0), t_1, s) \wedge \\
\quad TState(Do(Fire, t_1, t_2, s), t_1+1) = TState(s, t_1)
\end{array}
$$

Put in words, loading has the effect that the gun can be shot twice, spinning has the possible effect that a loaded gun becomes unloaded, and firing has the effect that, first, the number of possible shots is reduced by one and, then, that the turkey dies provided it was alive. Consider, for example, $S_{final} = Do(Fire, t_3, t_4, Do(Spin, t_1, t_2, S_0))$, then the axiomatization entails that $Holds(Shot(2), t_3, S_{final}) \rightarrow Holds(Shot(1), t_4, S_{final}) \wedge \neg Holds(Alive, t_4, S_{final})$, given that $t_4 > t_3 > t_2 > t_1$.

Translating $\mathcal{K}$-IA Chronicles into the Fluent Calculus

The basic concept of features in the Features-and-Fluents framework corresponds to fluents in the fluent calculus, and an occurrence symbol corresponds to a function symbol of sort ACTION. In what follows, we present a translation from arbitrary chronicles in the $\mathcal{K}$-**IA** class into a set of axioms of the extended fluent calculus.

Fluents

Every (multi-valued) n-ary feature $F(\vec{x})$ is translated into an $n{+}1$-ary fluent $F(\vec{x}, y)$ in the fluent calculus, where the last argument y denotes the value of the feature in a state. We then use the following notation:

$$Holds(F(\vec{x}) \widehat{=} y, t, s) \stackrel{\text{def}}{=} Holds(F(\vec{x}, y), t, s)$$

This is accompanied by two domain constraints which stipulate that in every situation at every time the fluent has a unique value. Formally,

$$
\begin{array}{l}
(\exists y)\, Holds(F(\vec{x}, y), t, s) \wedge \\
{[Holds(F(\vec{x}, y_1), t, s) \wedge Holds(F(\vec{x}, y_2), t, s)} \\
\qquad\qquad\qquad\qquad \rightarrow y_1 = y_2]
\end{array}
\tag{5}
$$

The special case of binary features can be directly translated into fluents without the additional argument, which are then either true or false in a state. An additional axiom expresses the unique-name-assumption for all fluent and action symbols occurring in a chronicle.

Precondition and state update axioms

Our translation into precondition and state update axioms makes use of the world description $\langle \texttt{Infl}, \texttt{Trajs} \rangle$. To begin with, the Features-and-Fluents semantics does not use explicit precondition axioms but rather conditional effects of actions. Hence, the action preconditions in the fluent calculus only encode the information concerning the duration of an action. First, we will give some auxiliary definitions.

Let r be a partial state and t a timepoint expression, then $assign(r, t, s)$ is a fluent calculus formula saying that the state in situation s at time t satisfies r:

$$assign(r, t, s) \stackrel{\text{def}}{=} \bigwedge [Holds(f_i \widehat{=} x_i, t, s) \mid (f_i : x_i) \in r]$$

For a trajectory ν and timepoint expressions ς and τ, the duration is formalized by the following macro, where ς is the starting point and τ is the termination point of an action occurrence:

$$texpr(\nu, \varsigma, \tau) \stackrel{\text{def}}{=} (\tau - \varsigma = |\nu|)$$

Here, $|\nu|$ denotes the number of elements within the trajectory. Finally, for an action α, timepoint expressions ς and τ, and a partial state r, the following macro defines the range of possible durations of the action in situation s:

$$
\begin{array}{l}
duration(\alpha, \varsigma, \tau, r, s) \stackrel{\text{def}}{=} \\
\quad assign(r, \tau, s) \rightarrow \bigvee_{\nu \in \texttt{Trajs}(\alpha, r)} texpr(\nu, \varsigma, \tau)
\end{array}
$$

Definition 6. *Let an action a occurring in a $\mathcal{K}$-**IA** chronicle be given. The action precondition axiom for a is*

$$
\begin{array}{l}
Poss(a, t_1, t_2, s) \equiv (start(s) \leq t_1 < t_2) \wedge \\
\qquad\qquad \bigwedge_{r \in \mathcal{R}_a} duration(a, t_1, t_2, r, s)
\end{array}
$$

where $\mathcal{R}_a$ is the set of partial states for which the trajectories for action a have been defined.

The precondition axioms for the Russian Turkey scenario in Example 2, for instance, are obtained when translating the world description shown in Figure 2.

The definition of the state update axioms for a chronicle requires some further auxiliary notation:

$$state(\{f_1 \widehat{=} x_1, \ldots, f_n \widehat{=} x_n\}) \stackrel{\text{def}}{=} f_1(x_1) \circ \ldots \circ f_n(x_n)$$

Let r and r' be two partial states which satisfy $dom(r') \subseteq dom(r)$,[3] then the sets of positively and negatively changed

[3]For a partial state r, $dom(r)$ denotes the set of features for which r contains an assignment.

fluents between r and r' are defined as

$$\textit{effects}^-(r,r') \stackrel{\text{def}}{=} \textit{state}(\{f_i \widehat{=} x_i \mid (f_i : x_i) \in r \wedge (f_i : y_i) \in r' \wedge x_i \neq y_i\})$$
$$\textit{effects}^+(r,r') \stackrel{\text{def}}{=} \textit{state}(\{f_i \widehat{=} y_i \mid (f_i : x_i) \in r \wedge (f_i : y_i) \in r' \wedge x_i \neq y_i\})$$

Using the changes between consecutive partial states in a trajectory, the latter is mapped onto a conjunction of fluent calculus formulas that represent precisely these change. Formally, let $\nu = \langle r_1, r_2 \ldots, r_{n-1}, r_n\rangle$ be a trajectory in $\langle \texttt{Infl}, \texttt{Trajs}\rangle$ for partial state r and action a, then

$$\begin{aligned}
&\textit{strans}(a,r,\nu,s) \stackrel{\text{def}}{=} \\
&\quad \textit{TState}(\textit{Do}(a,t_1,t_2,s),t_1+1) = \\
&\qquad \textit{TState}(s,t_1) - \textit{effects}^-(r,r_1) + \textit{effects}^+(r,r_1) \\
&\quad \wedge \\
&\quad \textit{TState}(\textit{Do}(a,t_1,t_2,s),t_1+2) = \\
&\qquad \textit{TState}(\textit{Do}(a,t_1,t_2,s),t_1+1) \\
&\qquad\quad - \textit{effects}^-(r_1,r_2) + \textit{effects}^+(r_1,r_2) \\
&\quad \wedge \ldots \wedge \\
&\quad \textit{TState}(\textit{Do}(a,t_1,t_2,s),t_1+n) = \\
&\qquad \textit{TState}(\textit{Do}(a,t_1,t_2,s),t_1+n-1) \\
&\qquad\quad - \textit{effects}^-(r_{n-1},r_n) + \textit{effects}^+(r_{n-1},r_n)
\end{aligned}$$

The full state update axiom for an action is then obtained by the possible trajectories and the possible partial starting states.

Definition 7. *Let an action a occurring in a $\mathcal{K}$-**IA** chronicle be given. The state update axiom for a is*

$$\begin{aligned}
&\textit{Poss}(a,t_1,t_2,s) \rightarrow \\
&\quad \bigvee_{r \in \mathcal{R}_a, \nu \in \texttt{Trajs}(a,r)} \big[\textit{assign}(r,t_1,s) \wedge \textit{texpr}(\nu,t_1,t_2) \\
&\qquad\qquad \wedge\, \textit{strans}(a,r,\nu,s)\big]
\end{aligned}$$

The state update axioms in Example 2, for instance, are the result of translating the trajectories in Figure 2 (and removing the *texpr*-part, which for all three actions follows from the precondition axioms).

Schedule

The set of action statements of a schedule is mapped into fluent calculus axioms that define the overall final situation as containing exactly all the action occurrences of the schedule. More specifically, let $\{[\varsigma_1,\tau_1]\varepsilon_1,\ldots,[\varsigma_n,\tau_n]\varepsilon_n\}$ be the set of the action statements in the schedule, then

$$\begin{aligned}
&S_{\textit{final}} = \textit{Do}(A_n,s_n,t_n,\ldots,\textit{Do}(A_1,s_1,t_1,S_0)\ldots) \\
&\wedge \{[s_1,t_1]A_1,\ldots,[s_n,t_n]A_n\} = \\
&\qquad \{[\varsigma_1,\tau_1]\varepsilon_1,\ldots,[\varsigma_n,\tau_n]\varepsilon_n\}
\end{aligned}$$

where the second conjunct denotes the logical formula that expresses the equality of the two finite sets. The following axiom ensures that this action sequence is indeed executable:

$$\begin{aligned}
&\textit{Poss}(A_1,s_1,t_1,S_0) \wedge \ldots \wedge \\
&\textit{Poss}(A_n,s_n,t_n,\textit{Do}(A_{n-1},s_{n-1},t_{n-1},\ldots, \\
&\qquad\qquad \textit{Do}(A_1,s_1,t_1,S_0)\ldots))
\end{aligned}$$

The timing statements in a schedule set are directly taken as axioms in the fluent calculus.

Observations

The translation of the observations in a $\mathcal{K}$-**IA**-chronicle is based on mapping an atomic formula ϕ of the form $f \widehat{=} \mathcal{X}$, where $\mathcal{X} = \{x_1,\ldots,x_n\}$, into the fluent calculus axiom

$$\begin{aligned}
&\textit{transAssign}(\phi,t,s) \stackrel{\text{def}}{=} \\
&\quad \textit{Holds}(f \widehat{=} x_1,t,s) \vee \ldots \vee \textit{Holds}(f \widehat{=} x_n,t,s)
\end{aligned}$$

The macro *transAssign* is straightforwardly generalized to compound fluent formulas. Every elementary formula of the form $[\tau]\phi$ is then translated into the fluent calculus axiom $\textit{transAssign}(\phi,\tau,S_{\textit{final}})$.

Example 2. *The schedule and observations in Figure 1 for our variant of the Russian Turkey scenario are translated into fluent calculus axioms which—due to the given total order of the action occurrences—can be logically simplified to*

$$\begin{aligned}
&S_{\textit{final}} = \textit{Do}(\textit{Fire},s_3,t_3, \\
&\qquad\qquad \textit{Do}(\textit{Spin},s_2,t_2, \\
&\qquad\qquad\quad \textit{Do}(\textit{Load},s_1,t_1,S_0))) \\
&\wedge\, \textit{Holds}(\textit{Alive},0,S_{\textit{final}}) \wedge \neg\textit{Holds}(\textit{Alive},t_3,S_{\textit{final}})
\end{aligned}$$

Taken together, the fluent calculus axioms resulting from the translation of the Russian Turkey chronicle of Figures 1 and 2 entail, e.g., that $\textit{Holds}(\textit{Shot}(2),t_2,S_{\textit{final}})$ and $\textit{Holds}(\textit{Shot}(1),t_3,S_{\textit{final}})$.

Soundness and Completeness

We have presented a translation that maps a given scenario description from the Features-and-Fluents semantics into fluent calculus axiomatizations. In this section, we prove that this translation is sound and complete. The main idea of the proof is to show a one-to-one correspondence between the intended models of a $\mathcal{K}$-**IA** chronicle Υ and the models for the fluent calculus axiomatization $\Omega(\Upsilon)$. More specifically, we will compare the possible histories for Υ and the interpretation of $\textit{TState}(S_{\textit{final}},t)$ in the models for the fluent calculus translation according to the following definition. Without loss of generality, we consider only those first-order interpretations of $\Omega(\Upsilon)$ which use the same object domain $\mathcal{O}$ as Υ.

Definition 8. *Let Υ be a $\mathcal{K}$-**IA** chronicle and $\Omega(\Upsilon)$ its translation into the fluent calculus. A valuation-history pair $\langle M,R\rangle$ for Υ is* equivalent *to an interpretation $\mathcal{I}$ for $\Omega(\Upsilon)$ iff*

$$M[t] = t^{\mathcal{I}} \text{ for all timepoint constants } t \quad (6)$$
$$M[c] = c^{\mathcal{I}} \text{ for all object constants } t \quad (7)$$
$$R(f,t) = v \text{ iff } \textit{Holds}(f \widehat{=} v,t,S_{\textit{final}})^{\mathcal{I}} \text{ for all } f,t \quad (8)$$

In order to prove soundness and completeness of the fluent calculus translation, we first show that the set of intended model for a chronicle Υ without observations is equivalent to the set of models for $\Omega(\Upsilon)$.

Theorem 1. *Let Υ be a $\mathcal{K}$-**IA** chronicle with* $\text{OBS} = \{\}$. *For every intended model $\langle M,R\rangle$ for Υ there exists an equivalent model $\mathcal{I}$ for $\Omega(\Upsilon)$ and vice versa.*

Proof.
" $\Leftarrow$ ": Let $\mathcal{I}$ be a model for $\Omega(\Upsilon)$ with

$$S_{\mathit{final}} = \mathit{Do}(A_n, s_n, t_n, \ldots, \mathit{Do}(A_1, s_1, t_1, S_0) \ldots)$$

Let M be a valuation that satisfies (6) and (7), then M is defined for every constant occurring in Υ and it satisfies all equations and inequations of the schedule set because $\mathcal{I}$ is a model for the translation of the schedule set. Therefore, M is a correct valuation wrt. the chronicle Υ. We inductively construct a history R^n that satisfies (8).

For the base case $n = 0$, we know that for all f there is a unique v such that $\mathit{Holds}(f \widehat{=} v, t, S_0)^{\mathcal{I}}$ for all t, because $\mathcal{I}$ is a model for (5) and (2). We can thus define $R^0(f,t) = v$ iff $\mathit{Holds}(f \widehat{=} v, t, S_0)$. Since there are no observations in Υ and no actions which can influence the state at timepoint 0, M and R^0 satisfy all conditions of Definition 3.

For the induction step, let

$$S_{n-1} = \mathit{Do}(A_{n-1}, s_{n-1}, t_{n-1}, \ldots, \\ \mathit{Do}(A_1, s_1, t_1, S_0) \ldots)$$

and suppose given a history R^{n-1} such that for all t, $R^{n-1}(f,t) = v$ iff $\mathit{Holds}(f \widehat{=} v, t, S_{n-1})^{\mathcal{I}}$. We then construct R^n as follows:

$$\begin{array}{l} R^n(f,t) = R^{n-1}(f,t)\,, \text{ for all } t \leq s_n \\ R^n(f,t) = v \text{ iff } \mathit{Holds}(f \widehat{=} v, t, S_{\mathit{final}})^{\mathcal{I}}\,, \\ \qquad\qquad\qquad\qquad \text{for all } s_n < t \leq t_n \\ R^n(f,t) = R^n(f,t_n)\,, \text{ for all } t > t_n \end{array}$$

From $\mathit{Poss}(A_n, s_n, t_n, S_{n-1})^{\mathcal{I}}$ and the precondition axiom in $\Omega(\Upsilon)$ according to Definition 6 it follows that there are a partial state r and a trajectory $\nu \in \mathtt{Trajs}(A_n, r)$ such that both R^n satisfies the assignments in r at time s_n and $t_n - s_n = |\nu|$. Moreover, from the state update axiom in $\Omega(\Upsilon)$ according to Definition 7 and Proposition 1 it follows that for some such partial state r and trajectory ν, history R^n evolves between $s_n{+}1$ and t_n according to ν. From Definition 3, it follows that valuation M along with history R^n is an intended model for Υ, which is equivalent to $\mathcal{I}$ by construction.

" $\Rightarrow$ ": Given an intended model for Υ, an equivalent model for $\Omega(\Upsilon)$ can be constructed in a similar way. □

This result can be lifted to chronicles with observations according to the following theorem, which shows that an observation ψ is true wrt. a valuation and history iff the translation of ψ is true in an equivalent model for the fluent calculus axiomatization.

Theorem 2. *Consider a valuation M and history R for a $\mathcal{K}$-**IA** chronicle Υ and an equivalent interpretation $\mathcal{I}$ for $\Omega(\Upsilon)$. Let ψ be a logical formula and $\Omega(\psi)$ its translation into the fluent calculus, then ψ holds in the intended model $\langle M, R \rangle$ iff $\Omega(\psi)^{\mathcal{I}}$.*

Proof. There are two forms of logical formulas.

- If ψ is an elementary formula $[\tau]\phi$, then $M[\tau] = \tau^{\mathcal{I}}$ by (6). Furthermore, if ϕ is a fluent formula $f \widehat{=} \mathcal{X}$, then

$$\begin{array}{ll} & [\tau]\phi \text{ is true wrt. } M \text{ and } R \\ \text{iff} & R(M[f], M[\tau]) \in \mathcal{X} \\ \text{iff} & \bigvee_{v \in \mathcal{X}} \mathit{Holds}(f \widehat{=} v, \tau, S_{\mathit{final}})^{\mathcal{I}} \\ \text{iff} & \mathit{transAssign}(\phi, \tau, S_{\mathit{final}})^{\mathcal{I}} \end{array}$$

  For compound fluent formulas ϕ, the claim follows by structural induction.
- If ψ is a relational formula among object constants or timepoints, the claim follows by (6) and (7).

If ψ is a compound logical formula, the claim follows by structural induction. □

Consequently, every element of the observation set OBS restricts the set of intended models in the same way it restricts the corresponding interpretations for $\Omega(\Upsilon)$. This shows that the translation is sound and complete for general $\mathcal{K}$-**IA** chronicles.

Discussion

We have presented a significant extension of the fluent calculus for modeling actions with durations and trajectories of changes. We have shown that the extended calculus can handle the full range of ontological phenomena in $\mathcal{K}$-**IA**, the broadest ontological class of the Features-and-Fluents framework that has been fully formalized in (Sandewall 1994). To this end, we have developed a provably correct translation function that maps arbitrary $\mathcal{K}$-**IA**-chronicles into fluent calculus axiomatizations. As a simple consequence of this result, the fluent calculus is now expressive enough to handle the entire test suite of example problems in (Sandewall 1994).

While the focus in this paper has been on the fluent calculus, our mapping from chronicles into classical logic should be adaptable to related solutions to the frame problem. In the situation calculus (McCarthy 1963; Reiter 1991), for example, this could be achieved by using a combination of nondeterministic successor state axioms (Lin 1996) and an explicit model of time (Reiter 1998). Likewise, the event calculus (Shanahan 1997), to mention another popular approach, could be extended to allow for modeling specific trajectories of changes for actions with durations. So doing should allow one to generalize existing assessments of these calculi, like the one given in (Kartha 1993), which are restricted to much narrower ontological classes.

Rooted in the fluent calculus, the programming language FLUX has been developed, which provides an efficient system for reasoning about actions based on state progression (Thielscher 2005a). We intend to integrate the extension of the fluent calculus presented in this paper so as to provide an automatic reasoning tool for the entire $\mathcal{K}$-**IA** class based on FLUX. Another direction of future work would be be to enrich the existing ontological taxonomy of the Features-and-Fluents semantics to capture additional aspects which are readily available in the fluent calculus, like sensing actions (Thielscher 2000b) or surprises (Thielscher 2001).

References

Hanks, S., and McDermott, D. 1987. Nonmonotonic logic and temporal projection. *Artificial Intelligence* 33(3):379–412.

Kartha, G. N. 1993. Soundness and completeness theorems for three formalizations of actions. In Bajcsy, R., ed., *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 724–729. Chambéry, France: Morgan Kaufmann.

Lin, F. 1996. Embracing causality in specifying the indeterminate effects of actions. In Clancey, B., and Weld, D., eds., *Proceedings of the AAAI National Conference on Artificial Intelligence*, 670–676. Portland, OR: MIT Press.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence* 4:463–502.

McCarthy, J. 1958. Programs with Common Sense. In *Proceedings of the Symposium on the Mechanization of Thought Processes*, volume 1, 77–84. Reprinted in: (McCarthy 1990).

McCarthy, J. 1963. *Situations and Actions and Causal Laws*. Stanford University, CA: Stanford Artificial Intelligence Project, Memo 2.

McCarthy, J. 1990. *Formalizing Common Sense*. Norwood, New Jersey: Ablex. (Edited by V. Lifschitz).

Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., ed., *Artificial Intelligence and Mathematical Theory of Computation*. Academic Press. 359–380.

Reiter, R. 1998. Sequential, temporal GOLOG. In Cohn, A. G.; Schubert, L. K.; and Shapiro, S. C., eds., *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 547–556. Trento, Italy: Morgan Kaufmann.

Sandewall, E. 1994. *Features and Fluents. The Representation of Knowledge about Dynamical Systems*. Oxford University Press.

Sandewall, E. 1998. Cognitive robotics logic and its metatheory: Features and fluents revisited. *Linköping Electronic Articles in Computer and Information Science*. URL: `www.ep.liu.se/ea/cis/1998/017`.

Shanahan, M. 1997. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. MIT Press.

Thielscher, M. 1994. An analysis of systematic approaches to reasoning about actions and change. In Jorrand, P., and Sgurev, V., eds., *International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA)*, 195–204. Sofia, Bulgaria: World Scientific.

Thielscher, M. 2000a. Nondeterministic actions in the fluent calculus: Disjunctive state update axioms. In Hölldobler, S., ed., *Intellectics and Computational Logic*. Kluwer Academic. 327–345.

Thielscher, M. 2000b. Representing the knowledge of a robot. In Cohn, A.; Giunchiglia, F.; and Selman, B., eds., *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 109–120. Breckenridge, CO: Morgan Kaufmann.

Thielscher, M. 2001. The qualification problem: A solution to the problem of anomalous models. *Artificial Intelligence* 131(1–2):1–37.

Thielscher, M. 2005a. FLUX: A logic programming method for reasoning agents. *Theory and Practice of Logic Programming* 5(4–5):533–565.

Thielscher, M. 2005b. *Reasoning Robots: The Art and Science of Programming Robotic Agents*, volume 33 of *Applied Logic Series*. Kluwer.