

# Sensing Inertial and Continuously-Changing World Features

Theodore Patkos and Dimitris Plexousakis

**Abstract** Knowledge and causality play an essential role in the attempt to achieve commonsense reasoning in cognitive robotics. As agents usually operate in dynamic and uncertain environments, they need to acquire information through sensing inertial aspects, such as the state of a door, and continuously changing aspects, such as the location of a moving object. In this paper, we extend an Event Calculus-based knowledge framework with a method for sensing world features of different types in a uniform and transparent to the agent manner. The approach results in the modeling of agents that remember and forget, a cognitive skill particularly suitable for the implementation of real-world applications.

## 1 Introduction

For decades, research in Artificial Intelligence is concerned with understanding and formalizing the properties of commonsense reasoning. Contemporary progress in cognitive robotics has led to the development of agents with significant reasoning capabilities, demonstrating their applicability in high-level logic-based robot control systems operating in real-life uncertain and highly dynamic environments (i.e., [3, 17, 14]). Formal theories for reasoning about action and change are now widely accepted, building on their expressiveness and the ability to verify the correctness properties of their specifications.

Autonomous agents that act in real-world conditions usually operate with incomplete information about the world they inhabit; therefore they need to acquire information and extend their knowledge base (KB) at execution time. In such settings, the temporal aspects of knowledge plays an essential role. Both ordinary and knowledge-producing (sense) actions need to be performed at specific time instants and information acquired may only be considered valid for limited time periods. The

---

Theodore Patkos and Dimitris Plexousakis  
Institute of Computer Science, FO.R.T.H., Heraklion, Greece e-mail: {patkos,dp}@ics.forth.gr

---

*Please use the following format when citing this chapter:*

Patkos, T. and Plexousakis, D., 2009, in IFIP International Federation for Information Processing, Volume 296; *Artificial Intelligence Applications and Innovations III*; Eds. Iliadis, L., Vlahavas, I., Bramer, M.; (Boston: Springer), pp. 379–388.

significance of reasoning about what agents know or do not know about the current world state and how this knowledge evolves over time has been acknowledged as highly critical for real-life implementations [1, 11].

In previous work we have developed a unified formal framework for reasoning about knowledge, action and time within the Event Calculus for dynamic and uncertain environments [9]. The current paper extends the above framework with an account of knowledge-producing actions for both inertial and continuously changing world features in a uniform manner. For instance, a moving robot may be able to sense parameters, such as the state of doors, the number of persons around it or its current position. Knowledge about door states can be preserved in its memory until some relevant event changes it, while knowledge about the other features should be considered invalid after a few moments or even at the next time instant. Still, it is important that the act of sensing can treat the different contingencies in a transparent to the robot fashion.

The contribution of this study is of both theoretical and practical interest. The proposed approach allows sensing of inertial and continuously changing properties of dynamic and uncertain domains in a uniform style, providing a level of abstraction to the design of an agent's cognitive behavior. Moreover, it results in the development of agents that are able not only to remember, but also to forget information, either to preserve consistency between the actual state of the world and the view they maintain in their KB or due to restrictions, such as limited resources, which pose critical constraints when considering real-life scenarios. Finally, the approach is based on a computationally feasible formal framework.

The paper proceeds as follows. We first provide an overview of relevant approaches and background material about the Event Calculus and the knowledge theory. In Section 3 we describe the sensing methodology for inertial relations and in Section 4 we explain how this scheme can also support continuously changing aspects. Section 5 illustrates its application on more complex domains. The paper concludes with remarks in Section 6.

## 2 Background and Related Work

This study builds on previous work in Knowledge Representation that is concerned with reasoning about actions and causality and has resulted in a family of *action theories* based on variations of the predicate calculus. These theories are formal frameworks for reasoning with action preconditions and effects and have been extended to handle a multitude of commonsense phenomena, such as effect ramifications, non-deterministic actions, qualifications etc. Intense research in the field has been devoted to studying the interaction between action and knowledge, where sensing is treated as a form of action, whose effects do not affect the state of the world but only the agent's mental state. Many of the related studies, though, concentrate on sensing inertial world parameters.

Scherl and Levesque [12] developed a theory of action and sensing within the Situation Calculus, providing a solution to the frame problem by adapting the standard possible worlds specification of epistemic logic to action theories, an approach first proposed by Moore [7]. The significance of an explicit representation of time has been acknowledged in [11], where the formalism has been extended with a treatment of concurrent actions and temporal knowledge. Working on the Fluent Calculus, Thielscher [15] provided a solution to the inferential frame problem for knowledge, along with an elaborate introduction of the notion of *ability* for an agent to achieve a goal. Nevertheless, in both frameworks, once knowledge is acquired it is preserved persistently in memory; it can be modified by relevant actions, but is never lost. The action language  $\mathcal{A}_k$  [5] permits information to be retracted from the set of facts known by an agent, as a result of actions that affect the world non-deterministically. Still, sensing continuously changing world aspects is not considered.

Moreover, these frameworks are computationally problematic, due to their dependence on the possible worlds model. Many recent approaches adopt alternative representations of knowledge that permit tractable reasoning in less expressive domains. Petrick and Levesque [10], for instance, define a combined action theory in the Situation Calculus for expressing knowledge of first-order formulae, based on the notion of *knowledge fluents* presented in [2] that treated knowledge change at a syntactical level. To achieve efficient reasoning, knowledge of disjunctions is assumed decomposable into knowledge of the individual components. Still, sensing is again limited to inertial parameters and knowledge is never retracted.

Our approach is based on a knowledge theory that treats knowledge fluents in a style similar to [10], which uses the Event Calculus as the underlying formalism. The Event Calculus is a widely adopted formalism for reasoning about action and change. It is a first-order calculus, which uses *events* to indicate changes in the environment and *fluents* to denote any time-varying property. Time is explicitly represented and reified in the language propositions. The formalism applies the *principle of inertia* to solve the frame problem, which captures the property that things tend to persist over time unless affected by some event; when released from inertia, a fluent may have a fluctuating truth value at each time instant (we further elucidate these concepts in successor sections). It also uses *circumscription* [4] to support non-monotonic reasoning. The Event Calculus defines predicates for expressing which fluents hold when (*HoldsAt*), what events happen (*Happens*), which their effects are (*Initiates*, *Terminates*, *Releases*) and whether a fluent is subject to the law of inertia or released from it (*ReleasedAt*).

A number of different dialects for the Event Calculus have been proposed summarized in [13] and [6]. For our proposed knowledge theory we have employed and extended the discrete time axiomatization (DEC), thoroughly described in [8]. The knowledge theory can be applied to domains involving incomplete knowledge about the initial state, knowledge-producing actions, actions that cause loss of knowledge and actions with context-dependent effects. The axiomatization is restricted to reasoning about fluent literals, assuming that disjunctive knowledge can be broken apart into the individual components. To simplify the presentation, we assume perfect sensors for the agent, i.e., the result of sensing is always correct.

<b>Knowledge and the Commonsense Law of Inertia</b>	
$ReleasedAt(Knows((-)f),t)$	(KT1)
<b>Knowledge Persistence</b>	
$HoldsAt(KP((-)f),t) \Rightarrow HoldsAt(Knows((-)f),t)$	(KT2)
<b>Events with Known Effects</b>	
$\bigwedge_{i=1}^P [HoldsAt(Knows((-)f_i),t) \wedge Happens(e,t) \Rightarrow Initiates(e, KP(f_{pos}), t)]$	(KT3.1)
$\bigwedge_{i=1}^P [HoldsAt(Knows((-)f_i),t) \wedge Happens(e,t) \Rightarrow Terminates(e, KP(\neg f_{pos}), t)]$	(KT3.2)
$\bigwedge_{j=1}^N [HoldsAt(Knows((-)f_j),t) \wedge Happens(e,t) \Rightarrow Initiates(e, KP(\neg f_{neg}), t)]$	(KT3.3)
$\bigwedge_{j=1}^N [HoldsAt(Knows((-)f_j),t) \wedge Happens(e,t) \Rightarrow Terminates(e, KP(f_{neg}), t)]$	(KT3.4)
<b>Knowledge-producing (sense) Actions</b>	
$Initiates(sense(f), KPw(f), t)$	(KT4)
<b>Events with non-deterministic Effects</b>	
$\bigvee_{i=1}^P [\neg HoldsAt(Kw(f_i),t)] \wedge \bigwedge_{i=1}^P [\neg HoldsAt(Knows(\neg(-)f_i),t)] \wedge \neg HoldsAt(Knows(f_{pos}),t) \wedge Happens(e,t) \Rightarrow Terminates(e, KPw(f_{pos}), t)$	(KT5.1)
$\bigvee_{j=1}^N [\neg HoldsAt(Kw(f_j),t)] \wedge \bigwedge_{j=1}^N [\neg HoldsAt(Knows(\neg(-)f_j),t)] \wedge \neg HoldsAt(Knows(\neg f_{neg}),t) \wedge Happens(e,t) \Rightarrow Terminates(e, KPw(f_{neg}), t)$	(KT5.2)
$\bigwedge_{k=1}^R [\neg HoldsAt(Knows(\neg(-)f_k),t)] \wedge Happens(e,t) \Rightarrow Terminates(e, KPw(f_{rel}), t)$	(KT5.3)
<b>State Constraints (Indirect Knowledge)</b>	
$HoldsAt(Kw(f),t) \Leftrightarrow$ <There exists some state constraint about $f$ that is known to be triggered>	(KT7)

**Fig. 1** The DEC Knowledge Theory axiomatization, where  $e$  denotes an arbitrary event,  $f_{pos}$ ,  $f_{neg}$ ,  $f_{rel}$  are positive, negative and non-deterministic effects respectively, and  $f_i$ ,  $f_j$ ,  $f_k$  are the corresponding effect's preconditions.

Figure 1 summarizes the foundational axioms of the theory. The  $KP$  fluent ("knows persistently") denotes inertial knowledge as a result of direct action effects. Knowledge can also be inferred indirectly from state constraints.  $Kw$  ("knows whether") is an abbreviation for  $Kw(f) \equiv Knows(f) \vee Knows(\neg f)$ , and similarly for  $KPw$  (for a comprehensive treatment see [9]). Event variables are represented by  $e$ , fluent variables by  $f$  and variables of the timepoint sort by  $t$ <sup>1</sup>. The theory assumes the occurrence of an event  $e$  having direct domain effects captured by positive, negative and release effect axioms of the following form:

$$\begin{aligned} & \bigwedge_{i=1}^P [(\neg)HoldsAt(f_i, t) \Rightarrow Initiates(e, f_{pos}, t)] \\ & \bigwedge_{j=1}^N [(\neg)HoldsAt(f_j, t) \Rightarrow Terminates(e, f_{neg}, t)] \\ & \bigwedge_{k=1}^R [(\neg)HoldsAt(f_k, t) \Rightarrow Releases(e, f_{rel}, t)] \end{aligned}$$

Such action occurrences can also cause knowledge effects to the agent's mental state, which are captured by the foundational axioms of our knowledge theory. For instance, axiom (KT3.1) states that whenever an agent knows all preconditions of an occurring event  $e$  that causes  $f_{pos}$  to become true, then it will also *know* the truth value of that effect.

<sup>1</sup> All free variables are implicitly universally quantified.

As a running example for the rest of the paper, we assume a robot named *Rob* either standing or moving with constant velocity  $v$  and able to sense the state of doors, the number of persons near him and its current location, represented by fluents  $Closed(door)$ ,  $PersonsNear(robot, num)$  and  $Position(robot, pos)$ , respectively. Initially, Rob is unaware of any relevant information, i.e.,  
 $\neg\exists d HoldsAt(Kw(Closed(d)), 0)$ ,  $\neg\exists n HoldsAt(Kw(PersonsNear(Rob, n)), 0)$  and  
 $\neg\exists p HoldsAt(Kw(Position(Rob, p)), 0)$ .

### 3 Inertial Fluents - Remembering and Forgetting

In general, a robot's descriptions of world states involve a large number of components that are assumed stable between action occurrences and preserve their properties for as long as occurring events do not affect them. A door remains open until some  $Close(door)$  event happens, while an object's color persists even when someone lifts it and moves it around. This insight is related to the well-known *frame problem* of action theories, which is concerned with specifying the non-effects of actions. In the Event Calculus, the commonsense law of inertia expresses that certain objects tend to stay in the same state, unless an event happens that changes this state. We refer to fluents that are subject to this law as inertial defined as follows:

**Definition 1.** A fluent is called *inertial* if it tends to maintain its truth value, unless affected by some event.

An inertial fluent, denoted in the Event Calculus by the expression  $\neg ReleasedAt(f, t)$ , is always subject to inertia. One can easily observe that whenever an agent senses an inertial fluent, the knowledge gained can be stored and preserved in its KB for as long as no event causes its invalidation. Sensing inertial fluents is a well-formalized task in most related epistemic action theories. In order to model real-life agents with limited resources though, the assumption of permanent knowledge preservation is too strong to accept. Apart from issues related to memory storage capacity, the rate of change in dynamic worlds quickly renders information out-of-date, forcing the agent to reconsider its knowledge about the state of certain objects that ideally could have remained unaltered.

In order to represent the "fading" validity of knowledge-producing actions, we introduce a new sense action that extends the one presented in Figure 1. This action is captured by the following axioms:

$$Happens(sense(f), t) \Rightarrow Happens(remember(f), t) \wedge Happens(forget(f), t + T(f)) \quad (KT4.1)$$

$$Initiates(remember(f), KPw(f), t) \quad (KT4.2)$$

$$\neg Happens(sense(f), t) \Rightarrow Terminates(forget(f), KPw(f), t) \quad (KT4.3)$$

where  $T(f)$  denotes a function that introduces a time delay dependent on  $f$ 's properties (certain fluents tend to change more often than others). The axiomatization expresses that whenever an agent senses a fluent, two internal to the agent actions occur that cause knowledge about the state of the fluent to be kept in the memory for a specific time window. The *remember* action produces the traditional sensing effect, while the *forget* action's effect is canceled if a *sense* action occurs concurrently (axiom (KT4.3) is called *negative canceling effect* axiom for that reason). Notice how the *KP* fluent forces knowledge to become true, according to (KT2).

The above axiomatization provides two alternatives for modeling knowledge-producing actions for inertial fluents. For the purpose of constructing a theoretical framework the desirable side-effect of unlimited memory persistence of fluents is achieved by retracting the instance of the *forget* action from (KT4.1). In addition, an agent may also be equipped with the mental ability to forget, an essential cognitive skill for practical commonsense reasoning, particularly suited for real-life implementations. Furthermore, one may also use the same axiomatization as a means of sensing continuously changing fluents, as explained below.

## 4 Non-Inertial and Functional Fluents

Most current logic-based approaches that study the interaction of knowledge and time focus on sensing and obtaining knowledge about inertial fluents. Still, this is hardly the case when reasoning in dynamically changing worlds. Next, we show how the aforementioned approach can also be applied to broader classes of situations. First, we elaborate on the characteristics of such situations.

In addition to inertial fluents there are also fluents that change their truth value in an arbitrary fashion at each time instant. The number of persons entering a building or the mails arriving daily at a mailbox are typical examples. Such fluents are usually applied in order to introduce uncertainty, as they give rise to several possible models, and can be defined as follows:

**Definition 2.** A fluent is called *non-inertial* if its truth value may change at each timepoint, regardless of occurring events.

A non-inertial fluent is always released from inertia and is represented in the Event Calculus by the predicate  $ReleasedAt(f, t)$ . A particular use for non-inertial fluents has been proposed by Shanahan [13] as random value generators in problems, such as tossing a coin, rolling a dice etc, naming them *determining fluents*, as they determine non-deterministically the value of other world aspects.

For the purposes of epistemic reasoning, sensing non-inertial fluents provides temporal knowledge that is only valid for one time unit, i.e., it only reflects what is known at the time of sensing, but not what will be known afterwards. Whenever Rob needs to reason about the number of persons around him, it must necessarily perform a new sense action to acquire this information; any previously obtained knowledge may not reflect the current situation. Still, there is a class of non-inertial fluents that

is far more interesting, because it expresses continuous change that follows a well-defined pattern. Such fluents are utilized to denote gradual change (or *processes*, according to Thielscher [16]), for instance to represent the height of a falling object, the position of a moving robot, the patience of a waiting person etc. We call this class of fluents functional non-inertial fluents:

**Definition 3.** A non-inertial fluent is called *functional* if its value changes gradually over time, following a predefined function.

In order to represent gradual change in the Event Calculus, we first need to release the involved fluent from inertia, so that its value is allowed to fluctuate, and then we can apply a state constraint to restrain the fluctuation, so that the fluent can exhibit a functional behavior. For example, to express the change in Rob's location (on a single axis) while moving with constant velocity, we apply the following state constraint concerning the *Position(robot, pos)* fluent:

$$\begin{aligned} & \text{HoldsAt}(\text{Position}(\text{Rob}, \text{pos}), t_1) \wedge t > 0 \Rightarrow \\ & \text{HoldsAt}(\text{Position}(\text{Rob}, \text{pos} + (v * t)), t_1 + t) \end{aligned} \quad (\text{SC})$$

It is straightforward to observe how axioms (KT4) can accommodate sensing non-inertial fluents (both ordinary and functional). One just needs to set  $T$  equal to one time unit. Regarding functional fluents in particular, although the value of the sensed fluent is subject to change, knowledge about future values can still be inferred. The application of an epistemic logic axiom system, such as S5, along with DEC knowledge theory axioms combines the narrative of actions and observations with the agent's cognitive ability. Specifically, the distribution axiom (K) below dictates how the available state constraints can contribute to knowledge inference:

$$\begin{aligned} & \text{HoldsAt}(\text{Knows}(f_1 \Rightarrow f_2), t) \Rightarrow \\ & (\text{HoldsAt}(\text{Knows}(f_1), t) \Rightarrow \text{HoldsAt}(\text{Knows}(f_2), t)) \end{aligned} \quad (\text{K})$$

**Example 1** The previous discussion illustrates how the problem of sensing the two non-inertial fluents *PersonsNear(robot, num)* and *Position(robot, pos)* can be addressed. Imagine that Rob performs *Happens(sense(PersonsNear(Rob, num)), 0)* and *Happens(sense(Position(Rob, pos)), 0)* at timepoint 0. By forming the parallel circumscription of the example's domain theory (no initial knowledge and the two event occurrences) along with Event Calculus, Knowledge Theory and uniqueness-of-names axioms, we can prove several propositions<sup>2</sup>. First, two, internal to the robot, events will be triggered for each fluent; a *remember* event at timepoint 0 and a *forget* event at timepoint 1. For the *PersonsNear* fluent it can also be proved that

$$\begin{aligned} \models & \exists x \text{HoldsAt}(\text{Knows}(\text{PersonsNear}(\text{Rob}, x)), 1) \wedge \\ & \neg \exists x \text{HoldsAt}(\text{Kw}(\text{PersonsNear}(\text{Rob}, x)), 2) \end{aligned} \quad (4.1)$$

<sup>2</sup> Sample code for the DEC Reasoner tool (<http://decreasoner.sourceforge.net/>) is available at <http://www.csd.uoc.gr/~patkos/deckt.htm>

due to (KT4.2) and (KT2) at timepoint 0 and (KT4.3), (KT2) and (KT7) at timepoint 1. The case is different for the robot's position:

$$\models \exists p \text{HoldsAt}(\text{Knows}(\text{Position}(\text{Rob}, p)), t) \quad (4.2)$$

for all  $t > 0$ . This holds true, because, although the *forget* action results in  $\neg \text{HoldsAt}(\text{KPw}(\text{Position}(\text{Rob}, \text{pos})), t)$  for  $t \geq 1$ , axiom (K) transforms (SC) into  $\text{HoldsAt}(\text{Knows}(\text{Position}(\text{Rob}, \text{pos})), t_1) \wedge t > 0 \Rightarrow \text{HoldsAt}(\text{Knows}(\text{Position}(\text{Rob}, \text{pos} + (v * t))), t_1 + t)$

Consequently, once Rob senses his position at some timepoint, he can infer future positions, without the need to perform further sense actions. The state constraint provides all future derivations, affecting knowledge through (KT7).

## 5 Context-dependent Inertia

We can now formalize complex domains that capture our commonsense knowledge of changing worlds, where fluents behave in an inertial or non-inertial manner according to context. For instance, a robot's location is regarded as a continuously changing entity only while the robot is moving; when it stands still, the location is subject to inertia. As a result only while the robot *knows* that it is not moving can knowledge about its location be stored persistently in its KB in the style described in Section 3. In general, for any fluent that presents such dual behavior, there usually is some other fluent (or conjunction of fluents) that regulates its compliance to the law of inertia at each time instant. For the  $\text{Position}(\text{robot}, \text{pos})$  fluent, for instance, there can be a  $\text{Moving}(\text{robot})$  fluent that determines the robot's motion state. Such *regulatory* fluents appear in the body of state constraints to ensure that inconsistency does not arise when inertia is restored. According to their truth state, the fluent that they regulate can either be subject to inertia and maintain its value or released from it in order to be subject to a state constraint. To integrate regulatory fluents in the theory, (KT4.3) must be extended to ensure that the agent does not forget a fluent when it knows that the latter is inertial and should be kept in the KB:

$$\neg \text{Happens}(\text{sense}(f), t) \wedge \neg \text{HoldsAt}(\text{Knows}(\neg f_{\text{rglr}}), t) \Rightarrow \text{Terminates}(\text{forget}(f), \text{KPw}(f), t) \quad (\text{KT4.3}')$$

where  $f_{\text{rglr}}$  is  $f$ 's regulatory fluent. Notice that even when the agent is not aware of  $f$ 's inertia state, i.e.,  $\neg \text{HoldsAt}(\text{Kw}(f_{\text{rglr}}), t)$ , the axiom fires and knowledge about  $f$  is lost, to avoid preserving knowledge that does not reflect the actual state.

**Example 2** Imagine that Rob's movement is controlled by actions  $\text{Start}(\text{robot})$  and  $\text{Stop}(\text{robot})$  with effect axioms:

$$\text{Initiates}(\text{Start}(\text{robot}), \text{Moving}(\text{robot}), t) \quad (5.1)$$



$$\text{Terminates}(\text{Stop}(\text{robot}), \text{Moving}(\text{robot}), t) \quad (5.2)$$

While the robot is on the move, its position must no more be subject to inertia:

$$\text{Releases}(\text{Start}(\text{robot}), \text{Position}(\text{robot}, \text{pos}), t) \quad (5.3)$$

$$\text{HoldsAt}(\text{Position}(\text{robot}, \text{pos}), t) \Rightarrow \text{Initiates}(\text{Stop}(\text{robot}), \text{Position}(\text{robot}), t) \quad (5.4)$$

In addition, the state constraint that determines the location as the robot is moving, with the *Moving* fluent playing the regulatory role, is axiomatized as follows:

$$\begin{aligned} & \text{HoldsAt}(\text{Moving}(\text{robot}), t_1) \wedge \text{HoldsAt}(\text{Position}(\text{robot}, \text{pos}), t_1) \wedge \\ & t > 0 \wedge \neg \exists t_2 (\text{Happens}(\text{Stop}(\text{robot}), t_2) \wedge t_1 < t_2 < t_1 + t) \Rightarrow \\ & \text{HoldsAt}(\text{Position}(\text{robot}, \text{pos} + (v * t)), t_1 + t) \end{aligned} \quad (5.5)$$

As a result, axiom (KT4.3') will be instantiated as:

$$\begin{aligned} & \neg \text{Happens}(\text{sense}(\text{Position}(\text{Rob}, \text{pos})), t) \wedge \\ & \neg \text{HoldsAt}(\text{Knows}(\neg \text{Moving}(\text{Rob})), t) \Rightarrow \\ & \text{Terminates}(\text{forget}(\text{Position}(\text{Rob}, \text{pos})), \text{KP}_w(\text{Position}(\text{Rob}, \text{pos})), t) \end{aligned} \quad (5.6)$$

In brief, (5.6) states that the position should be stored if Rob knows that he is not moving. If, on the other hand, the robot does not possess such knowledge (even if he is unaware of his current mobility state, due to a potential malfunction), the information acquired will be retracted one time instant after the sense action. In this case, future knowledge can be inferred only if some state constraint is available.

Moreover, it can be proved that if Rob knows initially *whether* he is moving, a single sense action is sufficient to provide knowledge about all future locations, regardless of any narrative of *Start* and *Stop* actions before or after sensing. And, most importantly, Rob does not need to consider his current state when sensing; the knowledge theory abstracts the reasoning process of determining knowledge evolution, regardless of whether the sensed fluent is inertial or continuously changing.

## 6 Conclusions

In this paper, we have described a logic-based epistemic framework that addresses a common task in cognitive robotics, namely the way sense actions can update an agent's mental state given world features of different type. Non-logical frameworks, based for instance on probabilistic or qualitative reasoning, have also been proposed leading to computationally more appealing approaches, suffering though from well-known weaknesses concerning openness and correctness verification issues.

The proposed solution extends previous logic-based approaches, as it investigates a broad range of fluent types. It also provides a level of abstraction to the agent's cognitive mechanism, disengaging it from explicitly distinguishing the nature of

sensed fluents; whether sensed information should be stored or not is automatically determined by the agent's current knowledge. Different scenarios have been modeled using an offline reasoner. Our intension is to test the efficiency of the framework in more demanding domains and simulate online executions.

**Acknowledgements** The authors wish to thank Dr. Nick Bassiliades for stimulating comments and interesting discussions.

## References

1. Chittaro, L., Montanari, A.: Temporal Representation and Reasoning in Artificial Intelligence: Issues and Approaches. *Annals of Mathematics and Artificial Intelligence* **28**(1-4), 47–106 (2000)
2. Demolombe, R., Pozos-Parra, M.: A simple and tractable extension of situation calculus to epistemic logic. 12th International Symposium on Methodologies for Intelligent Systems (ISMIS-00) pp. 515–524
3. Fritz, C., Baier, J.A., McIlraith, S.A.: ConGolog, Sin Trans: Compiling ConGolog into Basic Action Theories for Planning and Beyond. In: Proceedings International Conference on Principles of Knowledge Representation and Reasoning (KR), pp. 600–610. Australia (2008)
4. Lifschitz, V.: Circumscription. *Handbook of Logic in Artificial Intelligence and Logic Programming* **3**, 297–352 (1994)
5. Lobo, J., Mendez, G., Taylor, S.R.: Knowledge and the Action Description Language A. *Theory and Practice of Logic Programming (TPLP)* **1**(2), 129–184 (2001)
6. Miller, R., Shanahan, M.: Some Alternative Formulations of the Event Calculus. In: *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II*, pp. 452–490. Springer-Verlag, London, UK (2002)
7. Moore, R.C.: A formal theory of knowledge and action. In: *Formal Theories of the Commonsense World*, pp. 319–358. J. Hobbs, R. Moore (Eds.) (1985)
8. Mueller, E.: *Commonsense Reasoning*, 1st edn. Morgan Kaufmann (2006)
9. Patkos, T., Plexousakis, D.: A Theory of Action, Knowledge and Time in the Event Calculus. In: *SETN '08: Proceedings 5th Hellenic Conference on Artificial Intelligence*, pp. 226–238. Springer-Verlag, Berlin, Heidelberg (2008)
10. Petrick, R.P.A., Levesque, H.J.: Knowledge Equivalence in Combined Action Theories. In: *KR*, pp. 303–314 (2002)
11. Scherl, R.: Reasoning about the Interaction of Knowledge, Time and Concurrent Actions in the Situation Calculus. In: *Proceedings 18th International Conference on Artificial Intelligence (IJCAI)*, pp. 1091–1098 (2003)
12. Scherl, R.B., Levesque, H.J.: Knowledge, Action, and the Frame Problem. *Artificial Intelligence* **144**(1-2), 1–39 (2003)
13. Shanahan, M.: The Event Calculus Explained. *Artificial Intelligence Today* **1600**, 409–430 (1999)
14. Shanahan, M., Witkowski, M.: High-Level Robot Control through Logic. In: *ATAL '00: Proceedings 7th International Workshop on Intelligent Agents VII. Agent Theories Architectures and Languages*, pp. 104–121. Springer-Verlag, London, UK (2001)
15. Thielscher, M.: Representing the Knowledge of a Robot. In: A. Cohn, F. Giunchiglia, B. Selman (eds.) *Proceedings International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pp. 109–120. Morgan Kaufmann, Breckenridge, CO (2000)
16. Thielscher, M.: The Concurrent, Continuous Fluent Calculus. *Studia Logica* **67**(3), 315–331 (2001)
17. Thielscher, M.: FLUX: A Logic Programming Method for Reasoning Agents. *Theory and Practice of Logic Programming* **5**(4–5), 533–565 (2005)