Uncertainty Discourse and Applications

www.uda-reapress.com

Manag. Anal. Soc. Insights. Vol. 1, No. 1 (2024) 66–72.

Paper Type: Original Article

A Note on Situation Calculus

Antonios Paraskevas*

[1] University of Macedonia, School of Information Sciences, Department of Applied Informatics, Information Systems and e-Business Laboratory (ISeB),156, Egnatia Str., 54636, Thessaloniki, Greece; aparaskevas@uom.edu.gr.

Citation:

Received: 3 October 2023 *Revised: 8 March 2024* *Accepted: 19 May 2024*	Paraskevas, A. (2024). A note on situation calculus. *Uncertainty discourse and applications, 1*(1), 66-72.

Abstract

Situation calculus is a logical language for expressing change. Situations, actions, and fluents are the three core ideas of situation calculus. As agents perform actions, the dynamic environment changes from one situation to another. Fluents are functions that change with the situation and describe the effects of actions. They can be seen as properties of the world that come into existence when an action is initiated and disappear when another action ends. While situation calculus is powerful, it often struggles with complexity and verbosity when modeling dynamic systems, making it challenging to manage and reason about in large-scale settings. We propose using Labelled Transition Systems (LTS) to address these limitations. The LTS model, based on graph models of modal logic, offers a more concise and formal representation of system behaviors. The LTS-based method aims to provide a simpler and more intuitive framework for modeling dynamic settings, thereby improving system representation clarity and efficiency. It allows for higher scalability and more efficient verification and validation processes, which are critical in complex systems. Finally, the LTS model seeks to bridge the theoretical expressiveness of situation calculus with the practical requirements of system design and analysis.

Keywords: Situation calculus, LTS model, Modal logic, Knowledge representation.

1 | Introduction

According to McCarthy and Hayes [1], a situation is "the complete state of the universe at an instance of time". In situation calculus, describing a dynamic domain involves determining the actions agents can take and the fluents needed to represent the changes that occur in the environment. Fluents are properties of the world that may vary throughout time. They are expressed as predicates, and their truth value depends on the context. For example, at (robot, location) might represent the location of a robot. Actions are atomic operations that can alter the condition of the world. They are expressed as predicate words, with arguments corresponding to the entities participating in the action. For example, move (robot, location) may denote the activity of moving a robot to a specified area. Situations depict the status of the world at various moments in

Corresponding Author: aparaskevas@uom.edu.gr jfea

time. They are commonly written as terms with S_0 describing the original situation and do (a, s) signifying the situation created by executing action a in situation s.

In the simplest version of the situation calculus, each action is described by two axioms:

I. The possibility axiom. The possibility axiom deals with the notion of whether an action is possible to execute in a given situation. This axiom specifies the conditions under which an action can be performed.

II. The effect axiom. The effect axiom specifies how the state of the world changes after acting. It describes the transition from one situation to another as a result of executing an action.

While situation calculus offers a formal and expressive basis for reasoning about actions and changes in dynamic environments, having found application in a wide variety of both theoretical and practical works [2–10], its complexity, scalability concerns, and limits in dealing with uncertainty and constant change can present challenges in practical AI applications. Specifically, traditional situation calculus relies heavily on layered quantification and complex temporal reasoning. It can be time-consuming and inefficient when modeling and analyzing dynamic systems, particularly as their size and complexity grow.

Moreover, the Labeled Transition System (LTS)-based approach facilitates modularity and scalability, allowing complex systems to be broken down into manageable components. This modular nature supports more straightforward integration and modification, enabling the analysis to scale effectively to larger systems without introducing ambiguity. The inherent alignment of LTS models with formal verification techniques, such as model checking, further ensures that properties like safety and liveness can be rigorously verified, providing confidence in the system's behavior under various conditions.

In this paper, we propose an alternative formalism based on the LTS model to describe the basic axioms of situation calculus, which provides a powerful and versatile framework for modeling, analyzing, and reasoning about dynamic systems. Modal logic graph models inspire this model and offer clarity, formal rigor, and an intuitive graphical representation, making it easier for both humans and machines to reason. The following section provides a summary of the LTS model.

2| LTS-Model

LTSs are a fundamental concept in theoretical computer science and formal methods. They provide a mathematical framework for modeling and analyzing the behavior of concurrent and reactive systems. LTSs have been successfully applied in epistemic logic [11], providing a formal and intuitive framework for representing and analyzing the knowledge and beliefs of agents as they evolve through actions and observations. They have also been effectively studied in model checking [12], [13]. In model checking, LTSs are used to depict the behavior of a system as a collection of states and transitions. These transitions are labeled with actions or events that guide the system from one state to another. By capturing the system's structure and dynamics, LTSs offer a concise yet comprehensive model for reasoning about system behavior.

Let Act be the universal set of observable actions, and let τ denote a local action that is unobservable to a component's environment. An LTS M is a quadruple $\langle \mathcal{Q}, \mathcal{A}, \delta, q_0 \rangle$ where [14]:

I. $\mathcal{Q}$ is a finite set of states.

II. $\mathcal{A} \subseteq$ Act is the communicating alphabet of M.

III. $\delta \subseteq \mathcal{Q} \times \mathcal{A} \cup \{\tau\} \times \mathcal{Q}$ is a labeled transition relation.

IV. $q_0 \in \mathcal{Q}$ is the initial state.

This paper extends the classical notion of an LTS by integrating concepts from modal logic [15]. In this vein, we define the next components [16]:

A proposition, p. A truth function, $tr(p, q_i)$, associates the truth value, t_i, of p in each state, q_i, where t_i=Tr(p, q_i). If p is a crisp proposition, t_i can be either true (1) or false (0).

A target set, $T(p)$, is a collection of "target states." A target state, q_j, is one in which t_j equals 1. Thus, $T(p)$ represents the set of all states in which p is true. p defines the target set, or $T(p)$. If q_j is the target state, it fulfills $T(p)$. Thus, $T(p) = \{q_j | tr(p, q_j) = 1\}$ (*Fig. 1*).

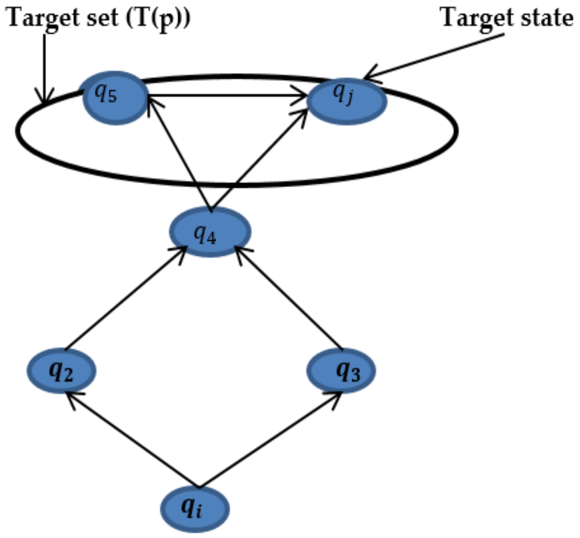

Fig. 1. Target set and target states.

Definition 1 ([16]). Each state, q_i, is connected with the set of all states, $R(q_i)$, that are accessible from q_i. $R(q_i)$ is known as the reachable set of q_i. Specifically, $R(q_i) = \{q_j | q_j \text{ is accessible from } q_i\}$ (*Fig. 2*).

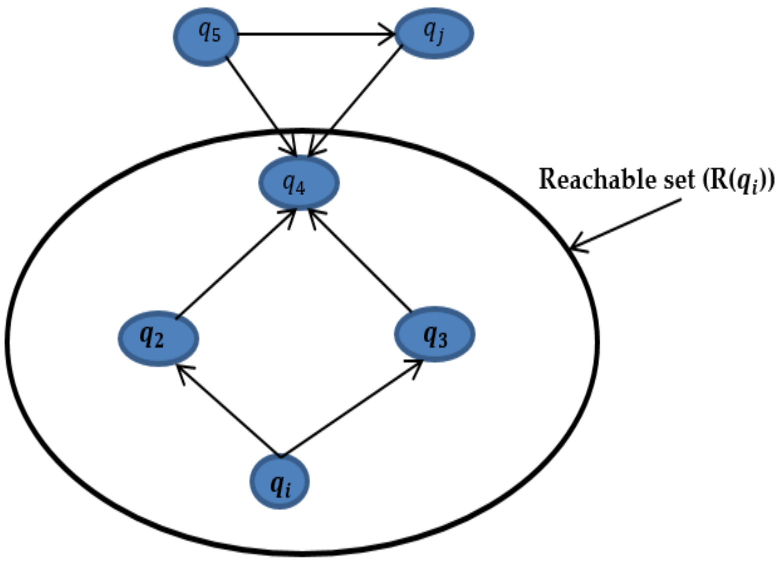

Fig. 2. Reachable set.

Definition 2 ([16]). p is possible in state q_i, abbreviated as possible p/q_i, if the intersection of $R(q_i)$ and $T(p)$ is not empty, that is, if there is a target state in $T(p)$ that can be reached from q_i.

3 | Axioms of Situation Calculus Using LTS-Model

3.1 | Possibility Axiom

The possibility axiom relates the existence of transitions labeled with actions to the possibility of performing those actions in the given states. In the context of modal logic, we can now redefine the possibility axiom used in situation calculus (Section 1) as follows:

Axiom 1. For a given proposition p and its associated truth function $tr(p, q_i)$, where q_i represents a state in the system

$$\text{Poss } (a,q) \Leftrightarrow \text{exists there } q_j \in T(p)\text{: tr } (a,q_j) = 1, \tag{1}$$

where poss(a, q) denotes the possibility of action a in state q.

T(p) represents the target set associated with proposition p, consisting of all states where p is true.

$tr\ (a, q_j)$ represents the truth value of action a in state q_j.

The axiom asserts that action a is possible in state q if and only if there exists a target state q_j where action a is true according to the truth function associated with p.

Alternatively, using definitions 1 and 2 from Section 2, we can reformulate the possibility axiom:

Axiom 2. For a given proposition p and its associated target set T(p), and a state q_i in the system

$$\text{Poss } (p, q_i) \Leftrightarrow R(q_i) \cap T(p) \neq 0, \tag{2}$$

where poss(p, q_i) denotes the possibility of proposition p in state q_i.

R(q_i) represents the set of all states reachable from state q_i.

T(p) represents the target set associated with proposition p, consisting of all states where p holds.

Now, the axiom states that proposition p is conceivable in state q_i if at least one state in $R(q_i)$ is also in T(p).

Example 1. To illustrate the expressiveness of our method, consider the following simple example taken from the world of an agent, which states that it is possible to shoot if the agent is alive and has an arrow [17]

$$\text{Alive(Agent, q)} \wedge \text{Have(Agent, Arrow, q)} \Rightarrow \text{poss(Shoot, q)}. \tag{3}$$

In LTS, transitions are labeled with actions, and states represent possible configurations of the system. Here's the simplified expression utilizing Eq. (1)

$$\text{Alive(q)} \wedge \text{Have(q, Arrow)} \Rightarrow \text{exists there } q_j \in T(q)\text{: tr(Shoot}, q_j) = 1. \tag{4}$$

The reformulated phrase asserts that if the agent is alive and possesses an arrow in state q, there exists a transition $q \to q_j$ in the set of possible transitions from state q, with the label indicating the possibility of shooting $(\text{tr(Shoot},q_j)=1)$.

The careful reader should notice that the if statement $q_j \in T(q)$: $\text{tr(Shoot},q_j)=1$ of the above expression is equal with the respective statement $R(q) \cap T(\text{Shoot}) \neq \emptyset$ as used in Eq. (2) of our conceptual framework, emphasizing the multiple facets of our approach that strengthen its expressiveness.

It is easy to see from the above that the LTS-model formulation, based on the possibility axiom, offers several advantages over classical situation calculus predicate logic. These benefits include conciseness, modularity, compatibility with formal techniques, reasoning efficiency, and simplicity of formalization.

3.2 | Effect Axiom

According to the effect axiom in situation calculus (see also Section 1), the effect of an action in a given state occurs if at least one state results from the action's execution and can be reached from the starting state.

Using the definitions used in our conceptual framework, we can define the aforementioned axiom as follows:

Axiom 2: For a given action a and its associated effect set $E(a)$, and a state q_i in the system

$$\text{Effect } (a, q_i) \Leftrightarrow \text{exists there } q_j \in R(q_i)\text{: } q_j \in E(a), \tag{5}$$

where Effect(a, q_i) denotes the effect of action a in state q_i.

$R(q_i)$ represents the set of all states reachable from state q_i.

$E(a)$ represents the effect set associated with action a, consisting of all states resulting from executing action a.

Our formulation of the effect axiom encapsulates the core of situation calculus's respective axiom, which states that the effect of an action in a given state occurs if there is at least one reachable state to which the action leads.

At this point, a more careful examination of the effect set $\mathrm{E(a)}$ is needed.

Definition 3. The effect set $\mathrm{E(a)}$ associated with an action is a collection of all states that occur from action a being performed from any state in the system. It can be defined as follows:

$$\mathrm{E(a)} = \bigcup_{\mathrm{q_i \in Q}} \mathrm{R(q_i, a)}, \tag{6}$$

where Q represents the set of all states in the system.

$\mathrm{R(q_i, a)}$ represents the set of all states reachable from state $\mathrm{q_i}$ through the execution of action a.

To put it simply, the effect set $\mathrm{E(a)}$ includes all potential states that might result from performing an action a from any state in the system. This concept defines the effect set $\mathrm{E(a)}$ as the potential changes in the system's state as a result of action a, while accounting for all possible beginning states $\mathrm{q_i}$.

This idea encompasses the full spectrum of possible outcomes resulting from the execution of an action, regardless of the initial condition. In other words, if we imagine the system as a network of interconnected states, where each state represents a different configuration of the system and an action leads to transitions between these states, the effect set includes all the states that can be reached through the action from any given state. The effect set is crucial for analyzing and verifying dynamic systems because it thoroughly examines all potential state transitions caused by an action.

Example 2. Now, let us consider an example taken from related literature [17] expressed within the terms of the effect axiom in situation calculus and then reformed within our framework to understand the advantages our approach offers.

The relational Holding axiom states that an agent is holding some gold g after performing a possible action if and only if the action was a grab of g, or if the agent was already holding g and the action did not release it:

$$\begin{aligned}\mathrm{Poss(a, s)} \Rightarrow \Big(\mathrm{Holding}\big(\mathrm{Agent, g, Result(a, s)}\big) \Leftrightarrow \mathrm{a}\\ = \mathrm{Grab(g)} \vee \big(\mathrm{Holding(Agent, g, s)} \wedge \mathrm{a} \neq \mathrm{Release\ (g)}\big)\Big).\end{aligned} \tag{7}$$

Given our terminology (Eq. (5) and Eq. (6)), we can reformulate the above axiom:

$$\mathrm{Effect(a, Holding(Agent, g))} \Leftrightarrow \textbf{exists there } \mathrm{q_j} \in \mathrm{R(Holding(Agent, g))}: \mathrm{q_j} \in \mathrm{E(a)}. \tag{8}$$

We can observe that Eq. (8) conveys the same meaning as Eq. (7) but more concisely and understandably. It allows for a more compact representation of actions and their effects, reducing the need for complex logical constructs and quantifiers. Thus, our approach offers advantages in modularity and clarity by separating action effects from the domain model. It also promotes ease of formalization and compatibility with formal methods, ultimately improving readability and scalability compared to traditional situation calculus expressions.

4 | Conclusion

Based on predicate logic, situation calculus can be difficult to use when describing dynamic system behaviors because it relies on layered quantification and complex temporal reasoning. It leads to challenges in scaling and ambiguity when depicting actions and their outcomes.

On the other hand, employing an LTS model within the situation calculus framework could offer several advantages. Firstly, the LTS model can provide a systematic and formal explanation of action sequences and their consequences, thereby improving the clarity and precision of reasoning about dynamic systems.

Secondly, the LTS model allows for a straightforward examination of state transitions and action connections, making tasks like planning and verification easier within the situation calculus framework.

Furthermore, LTS approaches naturally allow for modularity and scalability. LTS facilitates system integration and change by separating complex systems into manageable components and representing them as state-transition diagrams. This modular approach enhances the ability to extend the analysis to larger systems while maintaining cohesion and clarity.

In summary, integrating an LTS model into the situation calculus framework enhances the descriptive, manageable, and verifiable nature of dynamic system behaviors. This integration surpasses the constraints of classical situation calculus by offering a more understandable, scalable, and formal framework for analyzing actions and their outcomes.

Funding

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability

No new data were created or analyzed in this study. Data sharing does not apply to this article.

Conflicts of Interest

The author has no conflicts of interest to declare relevant to this article's content.

No new data were created or analyzed in this study. Data sharing does not apply to this article.

References

[1] Mccarthy, j, & Hayes, P. . (1969). Some philosophical problems from standpoint of artificial intelligence. In *Readings in artificial intelligence* (pp. 463–502). Elsevier.

[2] Farinelli, A., Finzi, A., & Lukasiewicz, T. (2007). Team programming in golog under partial observability. *IJCAI international joint conference on artificial intelligence*, *6*(12), 2097–2102.

[3] Ferrein, A., Fritz, C., & Lakemeyer, G. (2005). Using golog for deliberation and team coordination in robotic soccer. *AI - artificial intelligence*, *19*(1), 24–31.

[4] Ghaderi, H., Levesque, H., & Lespérance, Y. (2007). Towards a logical theory of coordination and joint ability. *Proceedings of the international conference on autonomous agents* (pp. 544–546). Association for Computing Machinery. DOI: 10.1145/1329125.1329223

[5] Reiter, R. (1991). The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. *Artificial and mathematical theory of computation*, *3*, 359–380. DOI:10.1016/b978-0-12-450010-5.50026-8

[6] Sardina, S., De Giacomo, G., Lespérance, Y., & Levesque, H. J. (2004). On the semantics of deliberation in IndiGolog - From theory to implementation. *Annals of mathematics and artificial intelligence*, *41*(2–4), 259–299. DOI:10.1023/B:AMAI.0000031197.13122.aa

[7] Scherl, R. B. (2022). A situation-calculus model of knowledge and belief based on thinking about justifications. *CEUR workshop proceedings*, *3197*, 104–114.

[8] De Giacomo, G., Felli, P., Logan, B., Patrizi, F., & Sardiña, S. (2022). Situation calculus for controller synthesis in manufacturing systems with first-order state representation (extended abstract). *IJCAI international joint conference on artificial intelligence* (pp. 5722–5726). Elsevier. DOI: 10.24963/ijcai.2022/798

[9] De Giacomo, G., Ternovska, E., & Reiter, R. (2020). Non-terminating processes in the situation calculus. *Annals of mathematics and artificial intelligence*, *88*(5–6), 623–640. DOI:10.1007/s10472-019-09643-9

[10] Baier, J. A., & McIlraith, S. A. (2022). Knowledge-based programs as building blocks for planning. *Artificial intelligence*, *303*, 103634. DOI:10.1016/j.artint.2021.103634

[11] Knight, S., Mardare, R., & Panangaden, P. (2011). *An epistemic logic for labelled transition systems*. McGill University. https://www.cs.mcgill.ca/~prakash/Pubs/completeness_paper.pdf

[12] Tretmans, J. (2008). Model based testing with labelled transition systems. In *Lecture notes in computer science* (Vol. 4949 LNCS, pp. 1–38). Springer. DOI: 10.1007/978-3-540-78917-8_1

[13] Baier, C., & Katoen, J.-P. (2008). *Principles Of model checking* (Vol. 950). MIT Press.

[14] Giannakopoulou, D., & Magee, J. (2003). Fluent model checking for event-based systems. *Proceedings of the acm sigsoft symposium on the foundations of software engineering* (pp. 257–266). IEEE. DOI: 10.1145/940071.940106

[15] Hughes, G. E., & Cresswell, M. J. (2012). *A new introduction to modal logic*. Routledge.

[16] Zadeh, L. A. (2014). A note on modal logic and possibility theory. *Information sciences*, *279*, 908–913. DOI:10.1016/j.ins.2014.04.002

[17] Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach*. Pearson.