Logical Foundations of Dynamical Answer Set Programming

Arivd Becker[1]

[1]University of Potsdam, Institute of Computer Science, An der Bahn 2, 14476 Potsdam

1. Introduction

Answer Set Programming (ASP) has established itself as a powerful rule-based formalism for declarative problem solving. Its combination of expressive rule-based modeling and non-monotonic reasoning makes it suitable for knowledge representation. Consequently, ASP has been successfully applied to a wide range of domains. However, many real-world applications involve dynamic environments that are governed by actions occurring over time, producing delayed effects, and causal dependencies among fluents. Examples include robotic intra-logistics systems, where robots transport goods through a warehouse, and job-shop scheduling problems, where operations require specific durations and influence the availability of resources. In a plain ASP encoding, this behavior typically needs to be modeled by introducing additional time steps. Although such encodings are possible, they obscure the underlying domain description and make it difficult to express general concepts such as action duration, delayed effects, and temporal dependencies directly.

Dedicated dynamic representation formalisms address these challenges by providing language constructs specifically designed for reasoning about actions and change. Respective languagesprovide explicit syntactic and semantic support for actions, enabling the direct representation of state transitions and temporal evolution. This allows to focus on the proper modeling of the dynamic domain rather than on the technical details required to simulate temporal evolution. Consequently, these formalisms provide a more intuitive and declarative way of specifying dynamic systems.

However, integrating these modeling capabilities into ASP is not straightforward. Existing dynamic formalisms are typically based on different semantic foundations and are not directly compatible with the non-monotonic semantics underlying ASP. Therefore, the challenge is not only to introduce additional syntax for dynamic concepts, but to develop a principled integration that preserves the semantic properties that make ASP effective.

This thesis investigates the theoretical foundations for integrating expressive dynamic representation formalisms into ASP by aligning them with the semantics of Equilibrium Logic. The goal is to extend ASP with intuitive language constructs for modeling dynamic systems while preserving its declarative nature and answer set semantics.

2. Background

2.1. Monotonic dynamic formalisms

One of the earliest and most influential approaches for reasoning about action and change is the Situation Calculus [1], a first-order logical language widely used for representing dynamic systems and planning problems. Building upon this foundation, GOLOG [2] introduced a high-level logic programming language for specifying complex actions and procedural control. Its ability to represent structured programs involving sequencing, choice, iteration, and recursion made it particularly attractive for

Joint Proceedings of the Workshops and Doctoral Consortium of the 42nd International Conference on Logic Programming (ICLP 2026), co-located with the Federated Logic Conference (FLoC 2026).

arvidbecker@uni-potsdam.de (A. Becker)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-03

CEUR-WS.org/Vol-4311/DC_short_03.pdf

applications such as robotics and industrial processes. Several extensions have further introduced support for concurrency and quantitative temporal constraints [3]. A central characteristic of these approaches is the explicit separation between actions and state properties, as well as the representation of procedural knowledge.

Another major family of approaches is represented by Action Languages [4], which provide concise formalisms for describing dynamic domains through action effects and transition systems. Inspired by STRIPS [5], Action Languages distinguish between actions and fluents and incorporate inertia directly into their semantics. Modern variants extend these capabilities with support for concurrency, indirect effects, and non-deterministic behavior [4].

Dynamic systems can also be modeled using trace-based approaches. Linear Temporal Logic (LTL) [6] is widely used for specifying and verifying reactive systems, but its expressive limitations regarding procedural behavior motivated the development of richer formalisms such as Dynamic Logic (DL) [7]. DL combines state-based specifications with path expressions that describe possible executions, thereby capturing the distinction between states and actions at the language level. Linear Dynamic Logic (LDL) [8] captures this idea and provides an expressive formalism that subsumes LTL while retaining favorable computational properties that make practical implementations feasible. Temporal and dynamic formalisms have also been applied to planning and control. For example, LTL-based approaches have been used to guide planning, while combinations of temporal specifications with GOLOG-style procedural knowledge have enabled richer representations of planning preferences and execution constraints [9, 10, 11]. Extensions such as Metric Temporal Logic [12] and timed variants of LDL [13] additionally support quantitative timing constraints.

2.2. Dynamic ASP

ASP provides several advantages for representing dynamic domains, most notably through its non-monotonic semantics. Default negation offers an elegant solution to the frame problem by allowing fluents to persist in the absence of information indicating change. Consequently, ASP has been successfully applied to planning and dynamic knowledge representation [14, 15, 16]. Since traditional ASP represents dynamic behaviour by encoding states and transitions over discrete time steps rather than through explicit action constructs or global temporal operators, several extensions have been proposed that incorporate actions into the syntax and semantics of the language. A major line of research extends the intermediate logic of Here-and-There (HT) to temporal and dynamic settings, resulting in Temporal Here-and-There (THT) and Dynamic Here-and-There (DHT). Based on these foundations, Temporal Equilibrium Logic (TEL) [17] and Dynamic Equilibrium Logic (DEL) [18] integrate temporal and dynamic operators into equilibrium semantics while preserving the non-monotonic reasoning principles underlying ASP.

3. Research Goals

My research aims at laying the theoretical foundations for expressive extensions of ASP that provide suitable language constructs for modeling and solving dynamic problems with temporal and metric constraints. In particular, the goal is to integrate metric and dynamic representation formalisms into HT- and Equilibrium Logic, thereby obtaining a semantic basis for dynamic and temporal reasoning under stable-model semantics.

A central aspect of this research is the investigation of which logical features are required for adequately modeling dynamic domains. This includes the study of expressive constructs such as complex path expressions, Golog-style procedural descriptions, metric temporal operators within a non-monotonic framework. While several of these constructs can already be represented within extended HT and Equilibrium Logic frameworks, their interaction with equilibrium minimization is not yet fully understood. In particular, it remains to be clarified how minimization affects the semantics of procedural constraints, and which fragments of the language admit intuitive and computationally feasible modeling principles.

Another important objective is the systematic investigation of formal properties of HT and Equilibrium Logic that are especially relevant for modeling practice. This includes the study of modularity principles, elaboration tolerance, and equivalence notions that support the structured development and maintenance of dynamic theories. Likewise, the project investigates equivalence-preserving transformations that allow expressive language extensions to be reduced to simpler temporal core languages. Current translations are often highly complex, and one aim is to identify more transparent and practically useful simplifications.

Ultimately, the long-term objective is the development of an expressive action language that systematically incorporates the elaborated temporal, metric, and dynamic features within an equilibrium-logic-based semantics. Such a language should provide both high expressive power for modeling complex dynamic domains and a rigorous semantic foundation that supports reasoning tasks, equivalence analysis, and practical implementation techniques.

4. Research Status

Recently, the research has focused on extending and adapting DEL toward a framework that is better suited for reasoning about action and change and in particular for capturing the simultaneous execution of actions. In particular, Dynamic Equilibrium Logic for Action and Change (DEL^{af}) was redesigned to more closely follow the distinction between actions and fluents known from Dynamic Logic and Action Languages. Finally, it was shown how the complexity of the formalism can be reduced to a well studied fragment that we refer to as temporal logic programs. Accordingly, we can compute models of DEL^{af} in telingo.

5. Preliminary Results

Important steps toward expressive dynamic extensions of ASP have already been achieved within our research group. This led to the development of the ASP systems telingo and delingo, implementing TEL and DEL over finite traces. To additionally capture quantitative timing constraints within ASP, a metric extension of TEL was developed in which temporal operators are parameterized by intervals over integers [19]. The resulting Metric Temporal Equilibrium Logic provides the formal foundation for an ASP-based framework capable of expressing both qualitative and quantitative temporal constraints. For example, this provides a way to express that pressing the button causes the traffic light to switch to green within 40 seconds: $\Box(press \rightarrow \Diamond_{[0,40]} green)$.

Building on these results, the combination of dynamic and metric reasoning was further investigated through the exploration of Metric Dynamic Equilibrium Logic [20]. Although the syntax of Metric Dynamic Logic was slightly adapted in our approach, the central idea of equipping dynamic operators with metric intervals was preserved. This augments the modalities of DEL with temporal intervals, enabling, for example, the specification that the effects of complex actions occur within a given time interval: $\Box([press?; \tau + (wait?; \tau)^*]\, green)$.

6. Open Issues and Challenges

An important open issue concerns the role of strong negation in DEL-basedformalisms. For modeling purposes, strong negation has proven particularly useful, for instance by allowing compact representations of inertia rules. However, several variants and semantic treatments of strong negation exist in the literature [21, 22]. A systematic comparison of these approaches is necessary in order to identify the variants that are most suitable for expressive and intuitive modeling of dynamic domains.

Another central objective is the preservation of ASP's high expressive power with respect to first-order representations. In particular, the long-term goal is to extend the developed framework for reasoning about action and change to the first-order level.

Furthermore, an important research question concerns the identification of language fragments with favorable computational properties. Since the full formalism is expected to be highly expressive, it is necessary to determine which fragments admit efficient reasoning procedures and are suitable for practical implementation. This includes the analysis of computational complexity, equivalence-preserving simplifications, and restrictions that still retain sufficient expressive power for relevant modeling tasks. A first prototype implementation, based on meta-ASP techniques, is expected to serve as a basis for evaluating both the expressive capabilities and computational behavior of the framework.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] R. Reiter, Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems., MIT Press, 2001.

[2] H. Levesque, R. Reiter, Y. Lespérance, F. Lin, R. Scherl, GOLOG: A logic programming language for dynamic domains., Journal of Logic Programming 31 (1997) 59–83. doi:10.1016/S0743-1066(96)00121-5.

[3] T. Hofmann, G. Lakemeyer, A logic for specifying metric temporal constraints for golog programs, in: G. Steinbauer, A. Ferrein (Eds.), Proceedings of the Eleventh Workshop on Cognitive Robotics (CogRob'18), volume 2325, CEUR Workshop Proceedings, 2019, pp. 36–46. URL: http://ceur-ws.org/Vol-2325.

[4] M. Gelfond, V. Lifschitz, Action languages, Electronic Transactions on Artificial Intelligence 3 (1998) 193–210. URL: http://www.ep.liu.se/ej/etai/1998/007/.

[5] R. Fikes, N. Nilsson, STRIPS: A new approach to the application of theorem proving to problem solving, Artificial Intelligence 2 (1971) 189–208.

[6] A. Pnueli, The temporal logic of programs, in: Proceedings of the Eighteenth Symposium on Foundations of Computer Science (FOCS'77), IEEE Computer Society Press, 1977, pp. 46–57. doi:10.1109/SFCS.1977.32.

[7] D. Harel, D. Kozen, J. Tiuryn, Dynamic logic, in: D. Gabbay, F. Guenthner (Eds.), Handbook of Philosophical Logic, volume 4, Springer-Verlag, 2001, pp. 99–107.

[8] G. De Giacomo, M. Vardi, Linear temporal logic and linear dynamic logic on finite traces, in: F. Rossi (Ed.), Proceedings of the Twenty-third International Joint Conference on Artificial Intelligence (IJCAI'13), IJCAI/AAAI Press, 2013, pp. 854–860.

[9] F. Bacchus, F. Kabanza, Using temporal logics to express search control knowledge for planning, Artificial Intelligence 116 (2000) 123–191.

[10] J. Baier, C. Fritz, M. Bienvenu, S. McIlraith, Beyond classical planning: Procedural control knowledge and preferences in state-of-the-art planners, in: D. Fox, C. Gomes (Eds.), Proceedings of the Twenty-third National Conference on Artificial Intelligence (AAAI'08), AAAI Press, 2008, pp. 1509–1512.

[11] M. Bienvenu, C. Fritz, S. McIlraith, Specifying and computing preferred plans, Artificial Intelligence 175 (2011) 1308–1345.

[12] R. Koymans, Specifying real-time properties with metric temporal logic, Real-Time Systems 2 (1990) 255–299. doi:10.1007/BF01995674.

[13] D. Basin, B. Bhatt, S. Krstic, D. Traytel, Almost event-rate independent monitoring, Formal Methods in System Design 54 (2019) 449–478. doi:10.1007/s10703-018-00328-3.

[14] Y. Dimopoulos, B. Nebel, J. Köhler, Encoding planning problems in nonmonotonic logic programs, in: S. Steel, R. Alami (Eds.), Proceedings of the Fourth European Conference on Planning, volume

1348 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 1997, pp. 169–181. URL: citeseer.nj.nec.com/dimopoulos97encoding.html.

[15] V. Subrahmanian, C. Zaniolo, Relating stable models and AI planning domains, in: Proceedings of the Twelfth International Conference on Logic Programming, MIT Press, 1995, pp. 233–247.

[16] V. Lifschitz, Answer set programming and plan generation, Artificial Intelligence 138 (2002) 39–54.

[17] P. Cabalar, R. Kaminski, T. Schaub, A. Schuhmann, Temporal answer set programming on finite traces, Theory and Practice of Logic Programming 18 (2018) 406–420. doi:10.1017/S1471068418000297.

[18] P. Cabalar, M. Diéguez, F. Laferriere, T. Schaub, Implementing dynamic answer set programming over finite traces, in: [23], 2020, pp. 656–663. doi:10.3233/FAIA200151.

[19] A. Becker, P. Cabalar, M. Diéguez, T. Schaub, A. Schuhmann, Metric temporal answer set programming over timed traces, Theory and Practice of Logic Programming 24 (2024) 425–452. doi:10.1017/S1471068424000139.

[20] A. Becker, P. Cabalar, M. Diéguez, L. Fariñas del Cerro, T. Schaub, A. Schuhmann, Metric dynamic equilibrium logic, Journal of Applied Non-Classical Logics 33 (2023) 495–519. doi:10.1080/11663081.2023.2244365.

[21] F. Aguado, P. Cabalar, J. Fandinno, D. Pearce, G. Pérez, C. Vidal, Revisiting explicit negation in answer set programming, Theory and Practice of Logic Programming 19 (2019) 908–924.

[22] F. Aguado, P. Cabalar, J. Fandinno, G. Pérez, C. Vidal, Explicit negation in linear-dynamic equilibrium logic, in: [23], 2020, pp. 569–576.

[23] G. De Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, J. Lang (Eds.), Proceedings of the Twenty-fourth European Conference on Artificial Intelligence (ECAI'20), IOS Press, 2020.