LUMI Agents: A Fuzzy BDI Framework for Intelligent Agents⋆

Muhammad Asaf[1,2,*], Paola Guarasci[2,3], Domenico Laurito[2], Davide Mario Longo[1], Angelo Mendicelli[2] and Francesco Scarcello[1]

[1]DIMES, University of Calabria, 87036 Rende (CS), Italy
[2]Artémat s.r.l., Via P. Bucci, Cubo 41C, 87036 Rende (CS), Italy
[3]DEMACS, University of Calabria, 87036 Rende (CS), Italy

Abstract

Autonomous agents must operate effectively in uncertain, dynamic environments that demand both symbolic reasoning and flexible adaptation. While classical BDI agents offer transparency and goal-directed behavior, they struggle with ambiguity and rely on predefined plans. Conversely, recent neural agents powered by Large Language Models (LLMs) demonstrate flexible planning capabilities but often lack structure, interpretability, and robust execution control. We introduce **LUMI** (Logical Uncertainty Modeling for Intelligent Agents), a unified architecture that integrates fuzzy reasoning and LLM-assisted planning. By grounding LLM reasoning within a fuzzy-symbolic architecture, LUMI provides symbolic control essential for developing green-aware agents, which must reason about the long-term impacts and trade-offs of their actions, in alignment with the objectives (and constraints) set by their designers. Its modular architecture enables cognitively grounded symbolic structures, fuzzy uncertainty modeling, and natural language-based task decomposition within a single decision-making pipeline.

LUMI is currently a work in progress. Here, we present a preliminary prototype with a use case in the domain of adaptive eLearning, where agents dynamically assess learner competencies, generate personalized learning plans, and adapt content in real time.

Keywords

Intelligent Agents, BDI Agent, Beliefs-Desires-Intentions, Fuzzy Logic, Fuzzy Behavior Tree (FBT), Large Language Models (LLMs), Multi-Agent Systems, Uncertainty Modeling

1. Introduction

As AI systems increasingly move into open, dynamic environments—from autonomous robots to adaptive learning tools—they are expected not only to perceive and act, but also to plan, reason, and adapt under uncertainty [1, 2]. Agents must operate in partially observable worlds, handle imprecise goals, and adjust their behavior in real time [1]. This is especially true for green-aware agents, which must reason about the long-term impacts and trade-offs of their actions in alignment with the objectives (and constraints) defined by their designers. However, most existing agent architectures are not designed to meet this level of complexity.

Symbolic models such as the Belief-Desire-Intention (BDI) architecture [3, 4] offer a cognitively grounded way to model rational behavior. They allow agents to maintain structured beliefs, pursue declarative goals, and commit to long-term intentions. Yet in practice, BDI agents typically rely on crisp logic and pre-defined plan-rules that limit their robustness in real-world scenarios.

To improve adaptability, researchers have extended agents to handle uncertainty, in particular by using fuzzy reasoning [5, 6] to evaluate goals with graded preferences, and reason under imprecision.

2nd Workshop on Green-Aware Artificial Intelligence, 28th European Conference on Artificial Intelligence (ECAI 2025), October 25–30, 2025, Bologna, Italy
*Corresponding author.
muhammad.asaf@dimes.unical.it (M. Asaf); paola.guarasci@unical.it (P. Guarasci); laurito@artemat.it (D. Laurito); davidemario.longo@unical.it (D. M. Longo); mendicelli@artemat.it (A. Mendicelli); scarcello@dimes.unical.it (F. Scarcello)
0009-0008-6116-8823 (M. Asaf); 0009-0005-8763-1431 (P. Guarasci); 0009-0009-9676-2950 (D. Laurito); 0000-0003-4018-4994 (D. M. Longo); 0009-0006-4206-3562 (A. Mendicelli); 0000-0001-7765-1563 (F. Scarcello)

Note that support for uncertainty has implications both for when some desire may be considered 'achieved' and for dealing with intentions.

At the same time, Behavior Trees (BTs) have gained popularity in robotics and game AI as an execution framework. However, BTs lack high-level cognitive modeling, restricting their expressiveness and responsiveness.

More recently, a completely new class of agents has emerged, built on the capabilities of Large Language Models (LLMs). Frameworks such as AutoGPT [7], ReAct [8], and Voyager [9] demonstrate how LLMs can generate goals, decompose tasks, and reflect on progress. These systems exhibit impressive generative flexibility but lack symbolic structure, leading to poor transparency, fragile execution, and limited goal-adherence over time.

No single paradigm seems to be sufficient on its own. Symbolic agents offer long-term planning adherence but struggle with uncertainty, while LLM-based agents generate creative plans but lack grounded reasoning and control. We argue that building green-aware agents that are both expressive and reliable requires something new that builds upon these previous contributions.

In this work-in-progress paper, we introduce **LUMI** (Logical Uncertainty Modeling for Intelligent Agents), a unified agent architecture that combines the structure of symbolic reasoning, the nuance of fuzzy control, and the flexibility of neural planning. A key motivation behind LUMI is the need to endow LLM-driven agents with explicit, logic-based control mechanisms. Unlike purely statistical approaches, LUMI enables agents not only to generate plans but also to evaluate and compare alternative strategies in a green-aware perspective. In this way, LUMI moves beyond feasibility and efficiency, supporting sustainable and responsible decision-making.

LUMI is a framework for creating and managing utility-driven agents inspired by the BDI paradigm, and it is based on

- **Fuzzy cognitive states**: beliefs and desires are modeled using fuzzy sets and utility functions, enabling reasoning under uncertainty;
- **Two-layer planning**: an *LLM-based planning layer* produces milestone-based strategies and suitable supporting agents to implement the low-level execution, guided and grounded by symbolic tools such as fuzzy reasoning modules and classical planners; a low-level planning based on a novel notion of *Fuzzy Behavior Trees (FBTs)*, which support continuous-valued action outcomes, dynamic replanning, and adaptive behavior.

These components work in concert to produce agents that are modular, interpretable, and robust to changing conditions. LUMI does not simply combine modules—it offers a coherent cognitive architecture that preserves symbolic structure while embracing uncertainty and open-ended reasoning.

We validate LUMI through a case study focused on intelligent self-learning in the computer engineering domain. The agent ingests unstructured documentation about the subject and the desired learning goals, and then it builds a fuzzy knowledge model, formulates personalized learning goals, and adapts exercises in real time to users' needs. This environment illustrates LUMI's capacity to integrate long-term reasoning, fuzzy execution, and neural planning in a practical, knowledge-rich setting.

The remainder of the paper is structured as follows: Section 2 reviews related work. Section 3 presents the LUMI architecture. Section 4 discusses the case study. Section 5 concludes with future directions.

2. Related Work

The Belief-Desire-Intention (BDI) model [3, 10], rooted in Bratman's theory of practical reasoning [11], remains a foundational paradigm for modeling rational agents. In this model, *beliefs* represent world knowledge, *desires* represent admissible goals, and *intentions* represent committed plans. This structure inspired the development of agent programming platforms such as 3APL [12], 2APL [13], Jason [14], and CAN [15], which incorporate symbolic reasoning and deliberative planning. However, these systems are largely based on crisp logic and deterministic control, making them less effective in dynamic or uncertain environments where agents must operate with partial knowledge and imprecise goals.

To address these limitations, researchers have extended the BDI paradigm with fuzzy logic [16], enabling agents to handle uncertainty, ambiguity, and graded preferences. Early efforts such as the AFDM model [17] integrated fuzzy reasoning into agent decision-making. Casali et al. introduced the notion of *graded BDI agents* [18, 19], later formalized as the *g-BDI* framework [5], which uses multicontext logic to model fuzzy beliefs, desires, and intentions. More recently, Cruz et al. [6] established a formal fuzzy modal logic to capture the semantics of fuzzy cognitive states. These systems provide a principled basis for imprecise reasoning, but they generally rely on static planning pipelines and lack tight integration with execution and planning layers.

BDI agents have also been applied in practical domains such as robotics and cyber-physical systems (CPS). For example, Wesz et al. [20] connected the Jason platform to the Robot Operating System (ROS) for reactive robotic agents, though without explicit fuzzy reasoning. Karaduman et al. [21, 22] developed fuzzy BDI agents for CPS environments, demonstrating effective adaptation in smart production systems, IoT applications, and heterogeneous sensor networks. Their agents outperformed classical models in responsiveness and scalability, with minimal computational overhead [23].

Other work has focused on integrating planning mechanisms into the BDI loop. Xu et al. [24] proposed a hybrid BDI architecture that incorporates classical planners to recover from intention failure and generate new plans dynamically. However, while these systems support symbolic planning or reactive execution, they do not handle uncertainty and do not leverage LLM-based decision-making.

Despite these advancements, most existing approaches treat reasoning, planning, and execution as separate components. Fuzzy BDI agents typically rely on predefined plans and lack integrated execution control. Reactive frameworks such as Behavior Trees (BTs) [25, 26] are modular and efficient, but remain disconnected from high-level goal management. Meanwhile, neural agents like AutoGPT [7], ReAct [8], and Voyager [9] showcase flexible planning using LLMs, but suffer from weak symbolic grounding, poor long-term consistency, and opaque decision-making.

Our work aims at overcoming these limitations by introducing a unified architecture—LUMI—that integrates fuzzy BDI reasoning, neural-assisted planning, and continuous-valued control in a cohesive and interpretable framework.

3. LUMI Overview

We next overview the structure of the framework by describing its core components: fuzzy cognitive modeling, utility-based goal selection, hybrid planning, and adaptive execution through the novel notion of Fuzzy Behavior Trees (FBTs).

As a running example, we consider a use case where a LUMI agent is designed to guide a student through the process of learning Java programming in a personalized and adaptive manner. Its overarching objective is to ensure that the student acquires the foundational and intermediate Java concepts at a sufficient level of mastery before advancing to more complex topics:

- it monitors the student's understanding and performance across several conceptual modules (e.g., Variables, OOP, Exception Handling, Collections, Concurrency);
- it uses fuzzy logic to assess the student's proficiency, handling partial, uncertain, or imprecise information;
- it adapts the learning path dynamically based on fuzzy reasoning, selecting suitable pedagogical actions such as recommending reviews, administering quizzes, or allowing progression.

3.1. Architecture

Figure 1 presents the high-level architecture. Central to LUMI is the Orchestrator, which manages the reasoning cycle and coordinates specialized agents: Evaluator (belief evaluation), Reasoner (fuzzy inference), Oracle (LLM-assisted planner), Crafter (execution planner), and co-agents responsible for validation (Auditors). Low-level execution is delegated to further agents specifically designed according to the current intentions, and named Sentinels, Executors, and Observers.

LUMI promotes modularity, interpretability, and robustness by decoupling cognitive functions while maintaining tight integration through control flow and fuzzy semantics. Each component operates with well-defined interfaces, supporting adaptive behavior in uncertain environments.

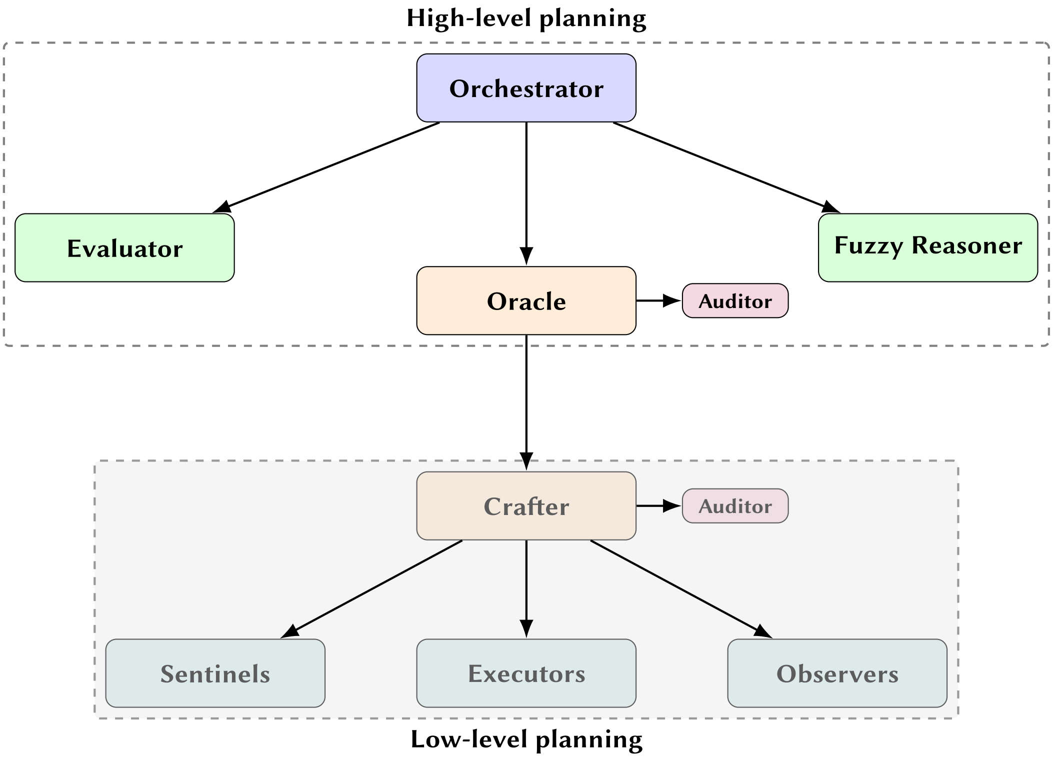

Figure 1: Two level planning in LUMI Architecture

3.2. Agent Roles and System Lifecycle

Each agent specializes a role. Table 1 summarizes responsibilities, from intention generation (Oracle) to fuzzy inference (Reasoner), planning (Crafter), and real-time sensing/action (Observers, Executors).

Table 1
LUMI Agents: Roles and Functions

Id	Agent	Roles and Functions
1	Orchestrator	Coordinates all actors and maintains the overall execution flow
2	Evaluator	Initializes and updates the Belief Knowledge Base (KB); assigns fuzzy degrees to facts; evaluates utilities
3	Fuzzy Reasoner	Manages fuzzy rules and constraints
4	Oracle	Generates intention plans based on beliefs and desires
5	Oracle Auditor	Verifies the proposed high-level plans
6	Crafter	Constructs FBTs for the selected intentions, and assigns tasks to execution nodes
7	Crafter Auditor	Verifies the proposed FBTs
8	Sentinels	Manage the logical flow in FBTs; monitor the execution progress
9	Executors	Perform concrete actions via APIs, tools, or internal routines; return fuzzy outcomes
10	Observers	Sense environment, evaluate fuzzy conditions, and update the belief base accordingly

Figure 2 shows the workflow among the components. LUMI's cycle proceeds from fuzzy evaluation $\rightarrow$ intention planning $\rightarrow$ FBT generation $\rightarrow$ execution $\rightarrow$ feedback.

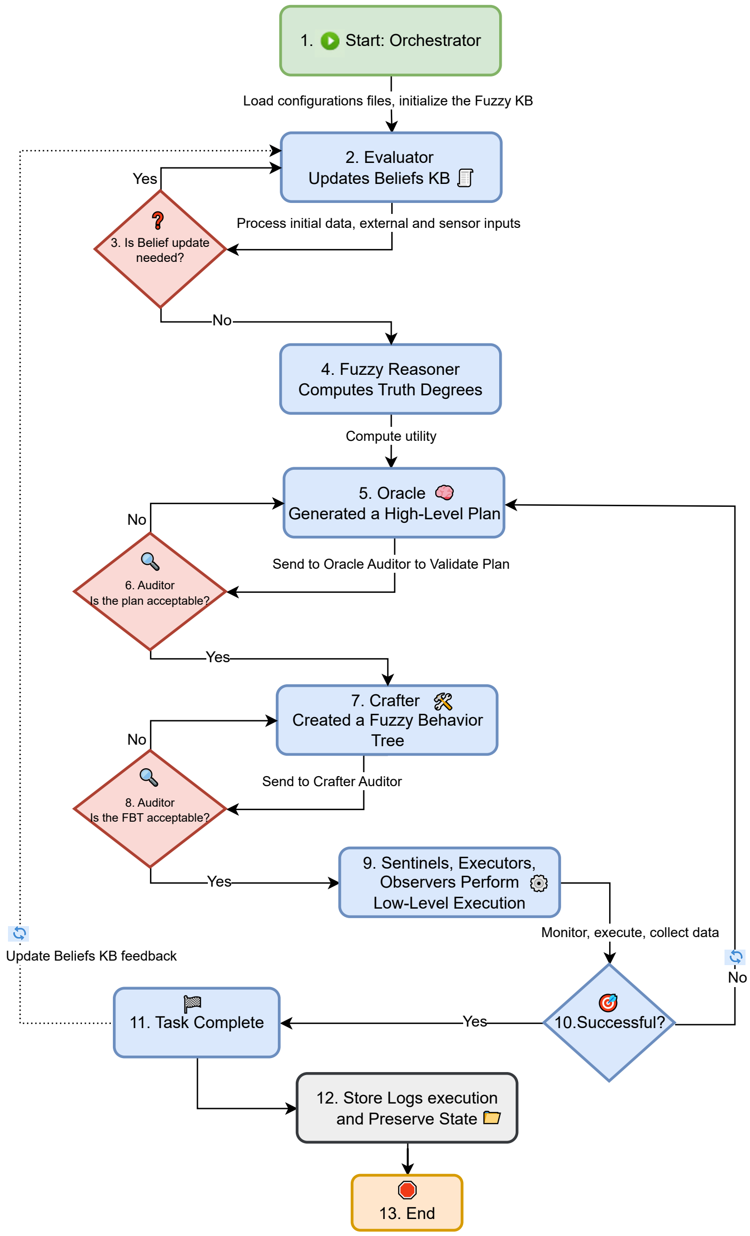

Figure 2: LUMI workflow (simplified for the case of one intention to be pursued)

3.3. Fuzzy Cognitive Layer

Beliefs represent the agent's understanding of the world using fuzzy sets theory. Based on the designer's input provided in a configuration file with possible attachments, the system builds and maintains through the Evaluator agent a knowledge base $\Sigma_B = \{a_1, ..., a_n\}$, whose elements are fuzzy atoms, each one equipped with a semantic descriptor and an optional witness set. The former is a natural language description of the role of the atom, which is useful for LLM agents, and the latter is an associated context extracted from the (optional) documents provided as part of the agent definition. At each state, beliefs are not used in rules in terms of binary values, but via fuzzy linguistic categories like *sufficient*, *moderate*, or *strong*. We assume hereafter basic knowledge of fuzzy logics [16].

Desires are expressed as a fuzzy formula that evaluates how well the current beliefs satisfy the agent goals, by considering the green-aware perspective, too. Note that this formula can be viewed as a utility function that the agent would like to maximize, until the output fuzzy value is good enough. In the running example, for the sake of simplicity, desires encode only curriculum-aligned objectives.

For instance, we can monitor the following notions encoded in the beliefs knowledge base:

- $\mu_{\text{OOP}}(x)$: proficiency in Object-Oriented Programming;
- $\mu_{\text{EH}}(x)$: proficiency in Exception Handling;
- $\mu_{\text{Collections}}(x)$: proficiency in Java Collections API.

Assume that fuzzy sets are defined over the interval $[0, 1]$ through suitable membership functions (defined according to the configuration file) that map values to levels of membership to the linguistic categories, for instance *Low*, *Medium*, and *High*.

For example, at a certain state, we may observe the following proficiency values, listed with their associated memberships to the linguistic categories:

$$\begin{aligned} \mu_{\text{OOP}}(x) = 0.7 \quad &\Rightarrow \text{high}(0.4),\ \text{medium}(0.6) \\ \mu_{\text{EH}}(x) = 0.6 \quad &\Rightarrow \text{high}(0.2),\ \text{medium}(0.8) \\ \mu_{\text{Collections}}(x) = 0.5 \quad &\Rightarrow \text{medium}(1.0) \end{aligned}$$

Note that an observed value for some property can be interpreted as belonging to multiple categories, with different degrees. In the example, the proficiency in Exception-handling is considered high with degree 0.2 and medium with degree 0.8.

This is an important property of fuzzy logic, where multiple rules can be activated simultaneously due to the overlapping nature of fuzzy membership functions. This allows different linguistic evaluations of belief atoms to co-exist and fire concurrently.

For instance the rules *If OOP is* high *and EH is* medium*, then recommend "Start Concurrency Module"* and the rule *If Collections is* medium *and EH is* high*, then recommend "Advanced Collections Exercises"*, should be evaluated together in the current state.

Assume that we use the *t-norm* $T(a, b) = a \cdot b$. Then, the former is evaluated as: $\text{Recc}_{SCM} = T(\mu_{\text{high}}(\text{OOP}), \mu_{\text{medium}}(\text{EH}))$; and the latter is evaluated as: $\text{Recc}_{ACE} = T(\mu_{\text{medium}}(\text{Collections}), \mu_{\text{high}}(\text{EH}))$. That is,

$$\mu_{\text{high}}(\text{OOP}) = 0.4, \mu_{\text{medium}}(\text{EH}) = 0.8 \quad \Rightarrow \quad \text{Recc}_{SCM} = 0.4 \cdot 0.8 = 0.32;$$

$$\mu_{\text{medium}}(\text{Collections}) = 1.0, \mu_{\text{high}}(\text{EH}) = 0.2 \quad \Rightarrow \quad \text{Recc}_{ACE} = 1.0 \cdot 0.2 = 0.2.$$

Based on these evaluations, the agent may choose to recommend *"Start Concurrency Module"* (or it may perform additional diagnostic actions to reduce uncertainty, if additional rules or policies encode such requirements).

Note that the rules in the knowledge base may also describe how the agent should behave, in particular by encoding green-aware rules (or constraints) defined by the user/designer.

In the current prototype, the *Fuzzy Reasoner* module handles only the classical modus-ponens–style rules. However, we are actively developing a more general and expressive fuzzy logic framework capable of supporting richer forms of inference.

3.4. Two-Layer Planning Architecture

The Oracle agent employs LLM-assisted reasoning to generate a high-level intention plan, which is subsequently reviewed by the Oracle Auditor to ensure compliance with all specified constraints. This plan constitutes a strategic blueprint that is interpreted as a collection of intentions designed to fulfill the agent's desires.

More precisely, the plan satisfies all predefined constraints and guarantees that, in any final state, the agent's beliefs satisfy the fuzzy formula encoding its desires.

Whenever an intention i is selected for execution, the Crafter agent generates a corresponding Fuzzy Behavior Tree (FBT), denoted T_i, which encodes a low-level plan for achieving i. This structure supports reasoning under conditions of partial satisfaction and enables real-time adaptation.

Formally, the FBT is defined as $T_i = (N, E, \lambda)$, where:

- N is a set of agents assigned to the roles of *Sentinels*, *Executors*, and *Observers*;
- E defines the tree structure through a set of directed edges;
- λ is a labeling function that assigns a fuzzy formula to each node.

Therefore, FBTs retain the expressive control patterns of BTs (sequence, fallback, parallel) while introducing fuzzy-valued leaf execution and evaluation nodes.

The Crafter Auditor is responsible for verifying the consistency and alignment between the generated plans and their corresponding execution structures.

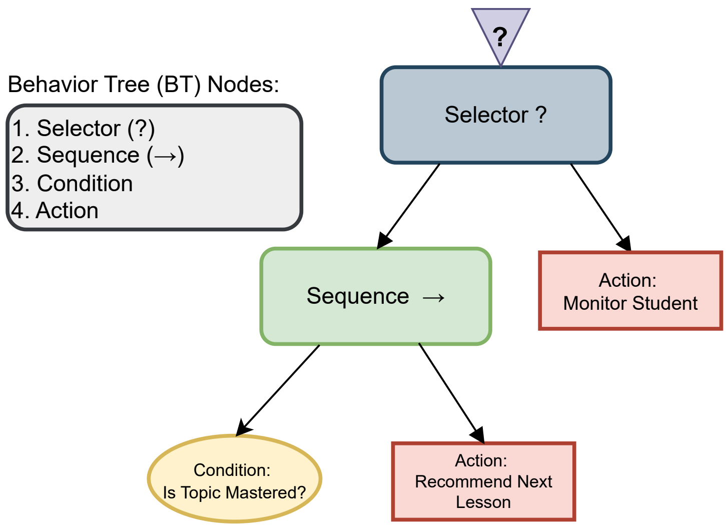

Figure 3: Behavior Tree (BT)

Figure 3 illustrates a fragment of an FBT. In contrast to standard `Sequence` nodes—which require all children to succeed—our framework evaluates these nodes using a chosen t-norm that aggregates the fuzzy satisfaction values of the children. Likewise, `Selector` nodes are evaluated using t-conorms, allowing for more flexible and graded assessments of success.

4. LUMI in Adaptive eLearning: the prototype

To validate the applicability and expressiveness of the LUMI framework, we look at real-world learning scenarios, to enable intelligent guidance, competency modeling, and real-time adaptation through dynamic agent coordination and fuzzy reasoning.

In fact, in contemporary digital learning environments, learners are often overwhelmed by unstructured resources, lacking coherent progression strategies. Our testbed focuses on a self-directed Java Enterprise learner.

The platform supports natural language interaction and external application integration through the Model Context Protocol (MCP). In the current prototypes, agents are executed via *AutoGen* and *LangGraph*, and use JSON message formats.

LUMI enables real-time adaptation with low-latency feedback. Some preliminary performance metrics are summarized in Table 2.

Table 2
Performance Evaluation of the LUMI

No.	Metric	Performance	Remarks
1	Session initialization	$<$ 5 seconds	Rapid startup and context loading
2	Belief update	$<$ 100 ms	Real-time feedback
3	Agent response time	2–5 seconds	Per-agent processing
4	Plan generation Time	2–4 minutes	Complex task planning

5. Conclusion

This paper has presented **LUMI**, a novel agent architecture that integrates fuzzy BDI reasoning, LLM-assisted planning, and fuzzy behavior trees into a unified framework for intelligent autonomous agents. LUMI extends classical BDI models with fuzzy logic to represent uncertainty in beliefs and preferences, while exploiting the generative and reasoning capabilities of Large Language Models to support abstract and open-ended planning. The introduction of *Fuzzy Behavior Trees* enables continuous, adaptive execution with fine-grained control under dynamic conditions.

A central goal of LUMI is to enable the development of green-aware agents that facilitate transparent, fair, and environmentally conscious decision-making across diverse domains such as finance, healthcare, and social services, where balancing sustainability and ethical considerations is increasingly crucial.

In this preliminary work, we explored LUMI's applicability in the context of adaptive e-learning for Java Enterprise development. In this use case, symbolic knowledge structures, fuzzy control, and neural planning are orchestrated across dynamically generated agents to guide learners, monitor competencies, and adapt content in real time.

Although promising, this initial implementation remains limited in scope. Future work should extend the framework by incorporating additional tools for LLM-driven agents and by supporting a more comprehensive fuzzy reasoning layer, which is the subject of an ongoing companion study.

Furthermore, the framework should be systematically evaluated by comparing LUMI with both classical BDI agents and LLM-based agents across established benchmark problems, to rigorously assess its effectiveness and unique contributions.

Declaration on Generative AI

The authors utilized ChatGPT and Grammarly to enhance language clarity and readability. The authors, who take full responsibility for the final version of the manuscript, carefully reviewed and refined all content generated by these tools.

6. Acknowledgments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9, Green-Aware AI, under the NRRP MUR program, funded by NextGenerationEU.

References

[1] M. Wooldridge, N. R. Jennings, Intelligent agents: Theory and practice, The knowledge engineering review 10 (1995) 115–152.
[2] N. R. Jennings, M. Wooldridge, Applications of intelligent agents, Agent technology: foundations, applications, and markets (1998) 3–28.
[3] A. S. Rao, M. P. Georgeff, et al., Bdi agents: from theory to practice., in: Icmas, volume 95, 1995, pp. 312–319.
[4] M. Wooldridge, An introduction to multiagent systems, John wiley & sons, 2009.

[5] A. Casali, L. Godo, C. Sierra, A graded bdi agent model to represent and reason about preferences, Artificial Intelligence 175 (2011) 1468–1478.
[6] A. Cruz, A. V. dos Santos, R. H. Santiago, B. Bedregal, A fuzzy semantic for bdi logic, Fuzzy Information and Engineering 13 (2021) 139–153.
[7] Significant Gravitas, AutoGPT: A Collection of Tools and Experimental Open-Source Attempts to Make GPT-4 Fully Autonomous, 2025. Accessed: 2025-06-14.
[8] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, in: International Conference on Learning Representations (ICLR), 2023.
[9] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, A. Anandkumar, Voyager: An open-ended embodied agent with large language models, arXiv preprint arXiv:2305.16291 (2023).
[10] A. S. Rao, M. P. Georgeff, Modeling rational agents within a bdi-architecture, Readings in agents (1997) 317–328.
[11] M. Bratman, Two faces of intention, The Philosophical Review 93 (1984) 375–405.
[12] K. V. Hindriks, F. S. De Boer, W. Van der Hoek, J.-J. C. Meyer, Agent programming in 3apl, Autonomous Agents and Multi-Agent Systems 2 (1999) 357–401.
[13] M. Winikoff, L. Padgham, J. Harland, J. Thangarajah, Declarative & procedural goals in intelligent agent systems, KR 2002 (2002) 470–481.
[14] M. Dastani, 2apl: a practical agent programming language, Autonomous agents and multi-agent systems 16 (2008) 214–248.
[15] R. H. Bordini, J. F. Hübner, M. Wooldridge, Programming multi-agent systems in AgentSpeak using Jason, volume 15, John Wiley & Sons, 2007.
[16] L. Zadeh, Fuzzy sets, Inf. and Control 8 (1965) 338–353.
[17] S. Shen, G. M. O'Hare, R. Collier, Decision-making of bdi agents, a fuzzy approach, in: The Fourth International Conference on Computer and Information Technology,, IEEE, 2004, pp. 1022–1027.
[18] A. Casali, L. Godo, C. Sierra, Graded bdi models for agent architectures, in: International Workshop on Computational Logic in Multi-Agent Systems, Springer, 2004, pp. 126–143.
[19] A. Casali, L. Godo, C. Sierra, g-bdi: A graded intensional agent model for practical reasoning, in: International Conference on Modeling Decisions for AI, Springer, 2009, pp. 5–20.
[20] R. B. Wesz, Integrating robot control into the Agentspeak (L) programming language, Master's thesis, Pontifícia Universidade Católica do Rio Grande do Sul, 2015.
[21] B. Karaduman, B. T. Tezel, M. Challenger, Enhancing bdi agents using fuzzy logic for cps and iot interoperability using the jaca platform, Symmetry 14 (2022) 1447.
[22] B. Karaduman, B. T. Tezel, M. Challenger, Rational software agents with the bdi reasoning model for cyber–physical systems, Engineering Applications of Artificial Intelligence 123 (2023) 106478.
[23] B. Karaduman, B. T. Tezel, M. Challenger, On the impact of fuzzy-logic based bdi agent model for cyber–physical systems, Expert Systems with Applications 238 (2024) 122265.
[24] M. Xu, T. Lumley, R. Fraga Pereira, F. Meneguzzi, A practical operational semantics for classical planning in bdi agents, in: ECAI 2024, IOS Press, 2024, pp. 1365–1372.
[25] M. Colledanchise, P. Ögren, Behavior trees in robotics and AI: An introduction, CRC Press, 2018.
[26] M. Iovino, E. Scukins, J. Styrud, P. Ögren, C. Smith, A survey of behavior trees in robotics and ai, Robotics and Autonomous Systems 154 (2022) 104096.