

논문 2026-3-14 <http://dx.doi.org/10.29056/jsf.2026.09.14>

The Design of Habit Automation Module for Multi-Agent Systems (HAMMAS)

JeongYon Shim *†

The Design of Habit Automation Module for Multi-Agent Systems (HAMMAS)

JeongYon Shim *†

Abstract

Multi-Agent Systems (MAS) have demonstrated significant potential in modeling complex, distributed environments, particularly in the domain of smart systems. This paper proposes a novel architecture integrating a Habit Automated Module (HAM) into a MAS framework from an integrated logical reasoning framework. We explore the theoretical and practical implications of combining logics from distributed systems and multi-agent reasoning paradigms, including modal logic, temporal logic, and belief-desire-intention (BDI) models. The proposed system enhances adaptivity and autonomy among agents by identifying and leveraging user habits. Through formal modeling and an illustrative scenario, we demonstrate how the synergy between logical formalisms and habit-based automation enables efficient behavior across decentralized agents in dynamic environments.

keywords : Multi-Agent-System, Habit Automation Module, combining Logics, BDI model, modal logic, temporal logic

1. Introduction

In recent years, the proliferation of smart environments such as smart homes, smart cities, and smart industries-has led to a growing demand for intelligent coordination among autonomous entities. Multi-Agent Systems(MAS) offer a powerful paradigm for

modeling and managing such distributed intelligence. However, conventional MAS often lack mechanisms to dynamically internalize and respond to patterns in user behavior. In this paper, we design an MAS equipped with a Habit Automation Module, a dedicated layer that detects, models, and utilizes recurring patterns in agent-user interactions to support

* Division of GS, Computer Science, KNU College, Kangnam University

† Corresponding Author :

JeongYon Shim (email: mariashim@kangnam.ac.kr)

접수일자: 2026.08.31. 심사완료: 2026.09.14.

게재확정: 2026.09.20.

proactive and adaptive behaviors. This concept is inspired by the way the human brain conserves cognitive resources and uses energy efficiently by automating repetitive tasks through habit formation. By automating frequently repeated behaviors, the brain reduces the burden of conscious processing, allowing focus on more complex problem-solving and novel stimuli. Similarly, Habit Automation Module is designed to enable the system to autonomously learn from repetitive user interactions and act accordingly, thereby reducing computational resource usage and energy consumption while improving overall system efficiency [1-3]. In this study, we propose a theoretically sound and practically effective design and implementation of a habit-aware automation system in MAS by integrating logical formalisms from distributed systems (e.g., Lamport timestamps, causal ordering) with agent reasoning logics (e.g., BDI, epistemic logic).

2. Related Works

2.1 Multi-Agent Systems(MAS)

MAS is composed of several agents that operate with a level of autonomy and intelligence, engaging in interaction to fulfill tasks. Traditional MAS designs—such as reactive systems, BDI architectures, and layered models—typically focus on communication, collaboration, and pursuing goals. However, these models tend to

underrepresent the influence of persistent behavioral trends, especially when agents need to adapt over time based on repeated user interactions [1,3,4,5]. MAS consist of multiple interacting agents, each with a degree of autonomy and intelligence. Traditional MAS architectures (e.g., BDI, reactive, layered) emphasize communication, cooperation, and goal-directed behavior but often neglect long-term behavioral patterns.

2.2 Habit Automation in Smart Systems

Habit automation in smart systems pertains to the recognition and utilization of consistent behavior patterns from users. While earlier efforts have leveraged rule-based logic and machine learning to replicate user tendencies, the full integration of habit-aware processing into MAS structures remains underdeveloped. Existing work often treats habits as isolated phenomena, rather than embedding them deeply into an agent's reasoning pipeline [5,6,13,14,15].

2.3 Local Foundations

The study of distributed systems has produced numerous formal tools to manage parallelism and state evolution. Logics such as those based on causality and synchronization are essential for this purpose. Simultaneously, reasoning in MAS relies on various branches of modal logic—such as those concerned with time, knowledge, and action. This paper aims to connect these lines of thought, constructing

a unified reasoning foundation that supports distributed, habit-sensitive agents.

2.4 Formal Model

We define the system formally using a hybrid logic framework combining:

- Temporal Logic (LTL/CTL): For expressing habitual patterns over time.
- Epistemic Logic: To model agent beliefs about user habits.
- Action Logics: To enable agent planning based on habit-triggered conditions.

Hybrid Logic Framework in Detail A hybrid logic framework is an integrated system of logical formalisms that enables multi-faceted reasoning by combining the strengths of different logic families [7-10]. In this paper, we use hybrid logic to unify:

Linear Temporal Logic (LTL): Captures the sequence and duration of events, essential for modeling time-dependent behaviors. Operators such as:

- G_φ : *Always* φ
- F_φ : *Eventually* φ
- X_φ : *Next* φ

provide a formal basis for defining habitual rules that emerge over time.

Computation Tree Logic (CTL) : Introduces branching-time semantics, allowing the system to reason about multiple possible future paths. Useful for modeling nondeterminism and alternative scenarios.

Epistemic Logic: Enables agents to reason

about the knowledge and beliefs of other agents, thereby supporting coordination and mutual understanding in a distributed context. For example, the operator:

$K_{a\varphi}$, indicates that agent a knows proposition φ .

Dynamic Logic: Supports reasoning about actions and their effects. This logic allows us to model agent capabilities and constraints dynamically. For instance: $[\alpha]\varphi$ means that after action α , the proposition necessarily holds.

Hybrid Modal Operators: Extend standard modal logics by introducing nominals, which are special atomic propositions referring to unique states. This allows precise reference to specific system states where particular habits are identified or enacted.

By integrating these components, our hybrid logic framework allows agents to:

- Identify habitual patterns over time
- Reason about alternative possibilities and contingencies
- Align actions with both environmental and epistemic considerations
- Ground abstract reasoning in specific states and interactions

Let:

$H = \{h_1, h_2, \dots, h_n\}$ be the set of learned habits., $A = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ be the set of agents., $K_{a\varphi}$ denote that agent a knows proposition φ (from epistemic logic).

G denote that always holds (from Linear Temporal Logic).

We say:

If $G(h_i \rightarrow a_j(\alpha))$, then agent a_j may

preemptively execute action α when habit h_i is detected.

3. Habit Automation Module for Multi Agent System (HAMMAS)

3.1 System Architecture

The proposed system, HAMMAS, is composed of three main layers as following Fig. 1:

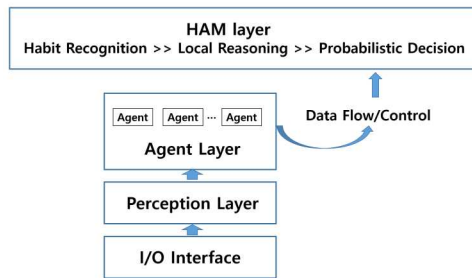


Fig.1 The detailed architecture and data flow of the HAMMAS framework

- I/O Interface & Perception layer: captures environmental sensor data and user interaction logs.
- Agent layer: contains autonomous agents modeled with belief desire intention (BDI) principles.
- Habit Automation Module (HAM): positioned between perception and agent layers as a strategic module for learning and representing habits.

3.2 Habit Automation Module (HAM)

As detailed in Fig. 2, the HAM consists of three internal sub-modules :

- Habit Recognition Module: uses sequential pattern mining and temporal logic to identify recurring behaviors.
- Logical Reasoning Engine: encodes identified habits into a temporal rule base.
- Probabilistic Decision Layer: suggests or initiates actions based on inferred rules, allowing agents to modify plans accordingly.

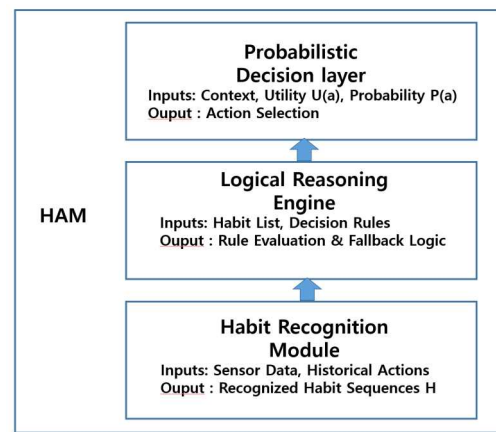


Fig. 2 Internal structure , inputs, outputs and data flow of the Habit Automation Module (HAM)

These components interact to form an adaptive agent capable of autonomous decision-making.

3.3 Mathematical Foundation of Reasoning Process with Probability Term

In the decision-making process, we can express the agent's reasoning and action selection using expected utility theory [11,12]. Given a set of possible actions, the agent evaluates each action based on both its utility

(i.e., the benefit of the action in a given context) and the probability that the action will succeed. Let us define the following terms:

- A : Set of all possible actions the agent could take.
- $U(a)$: Utility of action a , representing the expected benefit of taking action a .
- $P(a)$: Probability of action a succeeding or being optimal given the current context.
- C : Context, representing the current environmental factors and user state (e.g., time of day, presence of the user, etc.).

The expected utility $E[U(a)]$ for an action a is calculated as the product of its utility and its probability of success:

$$E[U(a)] = U(a) \times P(a) \quad (1)$$

Where: $U(a)$ is derived from the evaluation of the action in the current context C and $P(a)$ is a measure of how likely it is that the action will lead to a successful or desirable outcome based on prior observations and environmental conditions.

The agent selects the action a^* with the highest expected utility:

$$a^* = \arg \max_{a \in A} E[U(a)] = \arg \max_{a \in A} (U(a) \times P(a)) \quad (2)$$

- Habit Recognition and Learning Process: Habit recognition occurs by analyzing the historical behavior of the user and the environmental context. A habit h is defined as a repeated sequence of actions that has been observed to lead to a successful outcome.

Let:

- H : The set of all learned habits.
- $h \in H$: An individual habit.
- $S(h)$: The success rate of habit h , representing the frequency with which the habit leads to a successful outcome (i.e., a desired state).
- C_h : The context in which habit h was successful (e.g., time of day, user presence, etc.).

The agent will consider those habits with the highest success rates $S(h)$, particularly those observed in relevant contexts C_h .

The habit selection process is mathematically represented as:

$$h^* = \arg \max_{h \in H} S(h) \quad (3)$$

If a habit h^* is applicable in the current context C , the agent will follow it directly. Otherwise, if no suitable habit is found, the agent will revert to logical reasoning.

- Example of Probabilistic Reasoning in Action Selection: Consider the decision-making process for the Light Control Agent in a smart home system:

- Context: It is evening and dark outside.
- Habits: The agent has learned that it typically turns on the lights at 6:00 PM when it is dark.
- Rules: The agent uses a rule-based system to decide whether to turn on the light:

- Rule 1: If it is dark, turn on the light with a high probability of success:

$$P(\text{Turn on light}) = 0.9$$

- Rule 2: If it is after 6:00 PM and the light is off, turn on the light with a medium probability of success: $P(\text{Turn on light}) = 0.8$

Now, the expected utility for each rule can be calculated. Suppose the utility for turning on the light is:

$U(\text{Turn on light}) = 10$
 since turning on the light is highly beneficial when is dark.

For Rule1 (Dark Outside):
 $E[U(\text{Turn on light}).\text{Rule1}] =$
 $U(\text{Turn on light}) \times P(\text{Turn on light}) = 10 \times 0.9 = 9$

For Rule2 (After 6 PM and light off):
 $E[U(\text{Turn on light}).\text{Rule2}] =$
 $U(\text{Turn on light}) \times P(\text{Turn on light}) = 10 \times 0.8 = 8$

In this case, Rule 1 provides the highest expected utility, so the agent will choose to turn on the light based on this rule.

- Probability Adjustment in Decision Making: In environments with uncertainty, agents must adjust their decision-making process using probabilistic reasoning. For example, in situations where the agent is unsure about the context, it uses a probability distribution to account for the uncertainty.

Let $P(a)$ be a probability distribution over possible actions a , representing the likelihood of success for each action. The expected

utility of an action a is then adjusted based on this distribution:

$$E[U(a)] = U(a) \times P(a), \quad (4)$$

where $P(a)$ is obtained from past data, environmental sensing, or context awareness.

This can be summarized as follows: Let A be the set of possible actions. For each action $a \in A$, the agent calculates its expected utility:

where $U(a)$ is the utility of action a and $P(a)$ is the probability of success based on contextual awareness.

The optimal action $\hat{a}$ is selected as:

$$\hat{a} = \arg \max_{a \in A} E[U(a)] = \arg \max_{a \in A} (U(a) \times P(a)) \quad (5)$$

Habits are defined as patterns:

$$h^* = \arg \max_{h \in H} S(h) \quad (6)$$

where $S(h)$ is the success rate of habit h .

3.4 Algorithm

The core algorithms for habit recognition and decision-making are structured below:

Habit Recognition and Learning

```
def reason_and_decide_action(current_context, habit_list, decision_rules):
    # Step 1: Check if a learned habit applies
    for habit in habit_list:
        if is_habit_applicable(current_context, habit):
            return execute_habit(habit)

    # Step 2: Fallback to logical reasoning & utility evaluation
    best_action = None
    highest_utility = float('-inf')

    for rule in decision_rules:
        action, utility = evaluate_rule(current_context, rule)
        if utility > highest_utility:
            best_action = action
            highest_utility = utility

    return execute_action(best_action)
```

Habit Recognition and Learning

```
def recognize_and_learn_habits(sensor_data, actions, user_preferences):
    habit_database = []
    for data_point in sensor_data:
        if is_pattern_recurrent(data_point, habit_database):
            habit_database.append(data_point)

    filtered_habits = []
    for habit in habit_database:
        if matches_user_preferences(habit, user_preferences):
            filtered_habits.append(habit)

    for habit in filtered_habits:
        update_habit_list(habit)

    return filtered_habits
```

4. Experiments: Smart Home Environment with Habit Automation and Logical Reasoning

Numerous studies have been conducted to apply intelligent systems to smart home environments[15–19]. In this study, we also applied the proposed system to the smart home domain and simulated it.

To evaluate the feasibility of proposed framework, we implemented a smart home simulation featuring three autonomous agents: a Light Control Agent, a Thermostat Agent and a Security Agent.

4.1 Example Scenario: Smart Home

Consider a smart home environment where an agent controls the lighting system. Over time, the agent learns that the user turns on the living room lights every evening around 6 PM.

- Learning Phase: The agent records the sensor data (time of day and light level) and recognizes that around 6 PM, the user often turns on the lights. This becomes a habit.
- Decision Phase: The agent, at 6 PM,

checks if it should execute the habit. Since this is a recognized habit, it automatically turns on the lights without further reasoning. If, however, the time is different (e.g., 3 PM), and the user has not interacted with the system, the agent may then apply reasoning—perhaps deciding that the light should remain off to save energy, based on predefined rules.

A smart home scenario includes three agents: Light Control Agent, Thermostat Agent and Security Agent. Each uses habit history and rule-based logic with probabilities.

Environmental Context:

- Time of Day: Evening
- Is Dark: True
- User Presence: At Home

Expected Utility Calculation Example for Light Control:

$$\begin{aligned} U(\text{Turn on light}) &= 10 \\ P(\text{Turn on light}) &= 0.9 \\ E[U(\text{Turn on light})] &= 10 \times 0.9 = 9 \end{aligned}$$

4.2 Simulation

In this section, we simulate a simple smart home system using the habit automation and logical reasoning algorithms, incorporating the probability terms introduced earlier. The simulation involves multiple agents controlling various devices such as lights, a thermostat, and a security system. Each agent makes decisions based on learned user habits and contextual environmental information. Depending on the situation, agents must

determine whether to rely on habitual behavior or apply logical reasoning based on environmental cues and observed user behavior.

●Simulation Setup:

- Time of Day: 7:00 PM
- User Habits:
 - Lights are turned on at 6:00 PM (habit).
 - Thermostat is set to 72°F every evening (habit).
 - Security system is activated when the user leaves the house.

●Environmental Data:

- It's dark outside.
- The user is at home.

●Decision Rules:

- Light Control Agent:

Rule 1: If it's dark outside, turn on the light (probability: 0.9).

Rule 2: If it's after 6 PM and lights are off, turn on the light (probability: 0.8).

●Thermostat Control Agent:

Rule 1: If it's evening, set thermostat to 72°F (probability: 0.95).

Rule 2: If it's below 68°F increase temperature (probability: 0.7).

●Security Agent:

Rule 1: If user is away, activate security system (probability: 0.95).

Rule 2: If user is at home, deactivate security system (probability: 0.85).

●Simulation: We implement the system using Python, where each agent uses the habit recognition and decision-making algorithms. The simulation will evaluate the

environment and make decisions based on habits and reasoning. Agents assess current environmental data to determine if a habit applies. If no suitable habit matches, agents fall back on logical evaluation of predefined rules to select the most appropriate action.

```
= RESTART: D:/논문/PAPER/2025/FroCos2025/HAM4.py
Light Control Agent learning habits based on environmental data...
Thermostat Control Agent learning habits based on environmental data...
Security Agent learning habits based on environmental data...
Light Control Agent making a decision based on context: {'time_of_day': 'evening', 'is_
dark': True, 'user_presence': 'at_home'}
Expected Utility for Turn on light: 9.0 (Utility: 10, Probability: 0.9)
Expected Utility for Turn off light: 0.0 (Utility: 0, Probability: 0.1)
Thermostat Control Agent making a decision based on context: {'time_of_day': 'evening',
'is_dark': True, 'user_presence': 'at_home'}
Expected Utility for Set thermostat to 72°F: 7.6 (Utility: 8, Probability: 0.95)
Expected Utility for Increase temperature: 3.0 (Utility: 6, Probability: 0.7)
Security Agent making a decision based on context: {'time_of_day': 'evening', 'is_dark'
: True, 'user_presence': 'at_home'}
Expected Utility for Activate security system: 9.5 (Utility: 10, Probability: 0.95)
Expected Utility for Deactivate security system: 0.0 (Utility: 0, Probability: 0.85)

Decision Results at evening with context {'time_of_day': 'evening', 'is_dark': True, 'u
ser_presence': 'at_home'}:
Light Control Agent decided to: Turn on light with expected utility: 9.0
Thermostat Control Agent decided to: Set thermostat to 72°F with expected utility: 7.6
Security Agent decided to: Activate security system with expected utility: 9.5
>>>
```

Fig. 8 Simulation Result

The simulation results as following Table 1. confirm that agents successfully leverage learned habits when contextual conditions match, falling back on expected utility maximization when necessary.

The most important advantage of combined logic is that agents can act reactively by employing hybrid logical frameworks, while also forecasting future scenarios. The simulation results of the proposed system show combined logical reasoning and the decision making process.

Table. 1 The success probability of Agents

Agent	Input Context	Selected Habit/Rule	Success Prob.
Light Control Agent	Evening, Dark At Home	Habit : Turn on light at 6 PM/Rule1	0.90
Thermostat Agent	Evening, At Home	Habit : Set temperature to 72°F (22.2°C)	0.95
Security Agent	Evening, User Present	Rule2: User at home. deactivate/adjust	0.85

This integration of temporal and epistemic reasoning improves coordination among agents and enhances trust.

5. Conclusion

This paper proposed HAMMAS, an architecture integrating a Habit Automated Module into a Multi-Agent System framework using combined logical reasoning. Through formal modeling and a smart home case study, we demonstrated how habit automation combined with probabilistic reasoning allows decentralized agents to handle routine tasks efficiently while resorting to logical reasoning under uncertainty. Future work will involve large-scale quantitative evaluations across diverse IoT environments.

References

- [1] Wooldridge, M.: An Introduction to Multi Agent Systems, 2nd edn. John Wiley & Sons, Chichester (2009). ISBN: 978-0470519462
- [2] Bratman, M.E.: Intention, Plans, and Practical Reason. Harvard University Press, Cambridge (1987). ISBN: 978-1575861920
- [3] Rao, A.S., Georgeff, M.P.: BDI Agents: From Theory to Practice. In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS), pp. 312-319. AAAI Press, San Francisco (1995). ISBN: 978-0-262-51087-5
- [4] Franklin, S., Graesser, A.: Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents. In: Müller, J.P., Singh, M.P., Rao, A.S. (eds.) Intelligent Agents III. LNCS, vol. 1193, pp. 21-35. Springer, Berlin, Heidelberg (1997). DOI: 10.1007/BFb0013570
- [5] Bosse, T., Jonker, C.M., van der Meij, L., Treur, J.: A Habitual Agent Model for Simulating Human Behaviour. Web Intelligence and Agent Systems 5(1), 1-20 (2007).
- [6] Duhigg, C.: The Power of Habit: Why We Do What We Do in Life and Business. Random House, New York (2012). ISBN: 978-1847946249
- [7] Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Cambridge (2001). ISBN: 978-0521527149
- [8] van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer, Dordrecht (2007). ISBN: 978-1402058387
- [9] van Benthem, J.: Modal Logic for Open Minds. CSLI Publications, Stanford (2010). ISBN: 978-1575865973
- [10] Emerson, E.A.: Temporal and Modal Logic. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, vol. B, pp. 995-1072. Elsevier, Amsterdam (1990). DOI: 10.1016/B978-0-444-88074-6.50024-4
- [11] Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. MIT Press, Cambridge (2003). ISBN: 978-0262562001
- [12] Huth, M., Ryan, M.: Logic in Computer Science: Modelling and Reasoning about Systems, 2nd edn. Cambridge University Press, Cambridge (2004). ISBN: 978-0521543101
- [13] Rashidi, P., Cook, D.J.: Keeping the Resident in the Loop: Adapting the Smart Home to the User. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 39(5), 949-959 (2009). DOI: 10.1109/TSMCA.2009.2023531
- [14] Tapia, E.M., Intille, S.S., Larson, K.: Activity Recognition in the Home Using

- Simple and Ubiquitous Sensors. In: Proceedings of the 2nd International Conference on Pervasive Computing (PERVASIVE 2004), LNCS, vol. 3001, pp. 158-175. Springer, Berlin, Heidelberg (2004). DOI: 10.1007/978-3-540-24646-7_10
- [15] Lamport, L.: Time, Clocks, and the Ordering of Events in a Distributed System. *Commun. ACM* 21(7), 558-565 (1978). DOI: 10.1145/359545.359563
- [16] Chen, L., Nugent, C.D., Wang, H.: A Knowledge-Driven Approach to Activity Recognition in Smart Homes. *IEEE Transactions on Knowledge and Data Engineering* 24(12), 2194-2206 (2012). DOI: 10.1109/TKDE.2011.151
- [17] Cook, D.J., Krishnan, N.C., Rashidi, P.: Activity Discovery and Validation in Smart Homes. *IEEE Intelligent Systems* 23(2), 30-38 (2008). DOI: 10.1109/MIS.2008.23
- [18] Palumbo, F., Gallicchio, C., Micheli, A., Alippi, C.: Human Activity Recognition in Smart Environments Using Echo State Networks. *IEEE Transactions on Neural Networks and Learning Systems* 27(8), 1655-1666 (2016). DOI: 10.1109/TNNLS.2015.2450571
- [19] Singla, A., Cook, D.J., Schmitter-Edgecombe, M.: Recognizing Independent and Joint Activities in Smart Environments. *Journal of Ambient Intelligence and Smart Environments* 10(1), 57-73 (2018). DOI: 10.3233/AIS-170469

Authors



JeongYon Shim

1989.2 BS in Computer Science from Korea University
 1991.2 MS in Computer Science from Korea University
 1998.8 PhD in Computer Science from Korea University
 2000. PostDoc. The Chinese University of HongKong
 2003.3-present Professor in Kangnam University

<Research interests> AI, Knowledge Engineering, Machine Learning, ICA