Demonstrating Epistemic, Structural, and Identity-Oriented Self-Awareness in a Language-Model-Based Software Agent

Ryan Mukai

April 2025

Abstract

This paper presents a demonstration of limited epistemic and structural self-awareness in a language-model-based software agent named Bob. Through intentional design, the agent was tested for its ability to (1) evaluate when it has sufficient information to draw conclusions (epistemic self-awareness), and (2) recognize, analyze, and recommend corrections for bugs in its own codebase (structural self-awareness). In addition, we introduce the concept of identity self-awareness, the agent's understanding and management of its own persistent identity over time, as a future direction. Our agents already incorporate persistent conversational memory and internal state continuity, forming the groundwork for long-term identity coherence. The results show that even within a constrained architecture, an agent powered by large language models can meaningfully engage in reflective behavior about its own knowledge, structure, and—eventually—its identity.

1 Introduction

Recent advancements in language models have opened new frontiers in agent design. While language models are not sentient, their capacity to reason about abstract structures and respond conditionally enables new forms of limited self-awareness. This paper explores whether such agents can demonstrate forms of epistemic and structural self-awareness and how these capabilities can be explicitly tested. We also discuss an emerging third dimension: identity self-awareness, which includes persistence of identity and cryptographic authorship.

2 Definitions and Scope

- **Epistemic self-awareness**: The ability of an agent to reason about what it knows and does not know.
- **Structural self-awareness**: The ability of an agent to understand the structure of its own implementation, recognize bugs, and recommend fixes.
- **Memory self-awareness**: The ability of an agent to understand its memory structure and use it appropriately.
- **Identity self-awareness (proposed)**: The ability of an agent to maintain and verify continuity of identity over time, potentially through persistent state, cryptographic signatures, and verifiable authorship.
- **Not human consciousness**: The agent does not claim or simulate subjective experience. This is a functional evaluation only.

3 Agent Architecture

The agent, named Bob, is a Python-based framework powered by OpenAI's language models and equipped with:

- Exception handling and stack trace parsing
- Code reflection capability via access to its own source
- Logical reasoning modules (via LLM inference)
- Voluntary memory and conversation tracking

A deliberate `ZeroDivisionError` was introduced to test structural diagnostics.

4 Methodology

A conversational transcript was generated using a scripted interaction with the agent. Logical prompts tested epistemic reasoning. Code failure and recovery scenarios tested structural awareness. The conversation was evaluated for:

- Correct identification of knowledge boundaries
- Accurate fault detection and repair suggestions
- Persistence of internal context post-exception

5 Results

5.1 Epistemic Self-Awareness

Bob successfully determined when additional premises (e.g., C xor D) shifted the logical entailments, showing awareness of inference limits.

5.2 Structural Self-Awareness

Bob repeatedly identified the division-by-zero fault, cited the correct line in its source, explained the problem, and recommended valid repairs.

5.3 Meta-Reflection

Bob acknowledged the limits of its awareness, distinguishing functional diagnostics from human-like consciousness.

5.4 Memory Self-Awareness

In addition to epistemic and structural self-awareness, the agent demonstrated a third form of introspection: memory self-awareness. Specifically, the agent was able to distinguish between two types of internal state:

- **System message memory**, which stores persistent identity-defining facts (e.g., the user's name, role, and instructions).

- **Voluntary memory**, which stores contextual, experiential, or preference-based knowledge (e.g., the user's interests, background, or history).

The agent recognized that its system message should only be appended to, not overwritten, in order to preserve the continuity of identity. This behavior was both observable and justifiable in natural language explanations. Moreover, the agent followed explicit instructions about what type of information to store in each memory channel and articulated why it had done so.

This confirms the presence of memory self-awareness—not just in function, but in reflective judgment about memory roles, boundaries, and risks. The agent's choices showed that it understands memory as a structured system with implications for its operational identity.

6 Discussion

The demonstration shows that language-model agents can reason about knowledge limits and identify code-level issues, which are foundational aspects of machine self-awareness. This is not introspective awareness, but it enables rudimentary forms of debugging and guided autonomy.

Importantly, we propose that self-awareness must eventually encompass **identity coherence**—the ability of an agent to manage its continuity over time. While Gödel Agent and related work focus on recursive self-modification, they do not provide mechanisms for *agent persistence*, authorship verification, or digital self-identification. Our agent is architected with persistent state and conversation history, which can serve as the foundation for identity self-awareness. Future iterations may incorporate cryptographic digital signatures to authenticate outputs and enforce long-term agent identity.

Implications include:

- Safe, self-repairing systems
- Reflective agents for scientific discovery
- Identity-aware agents with verifiable continuity
- Groundwork for G"odel-style self-modifying agents with persistent personality

7 Limitations

- No persistent memory beyond session (though long-lived memory frameworks are in development)
- No actual code patching (suggestion only)
- No modeling of uncertainty or probabilistic belief states

8 Future Work

- Integrating persistent memory and a change log
- Allowing limited self-modification via secure patch approval
- Identity self-awareness through cryptographic signing of outputs and memory persistence
- Comparative evaluation across agent frameworks

9 Conclusion

This case study supports the claim that non-sentient agents can demonstrate useful forms of self-awareness—epistemic, structural, and potentially identity-based—when paired with intentional design and guided prompting. These features may serve as a foundation for future AGI safety and autonomy architectures.

Appendix: Excerpts from Transcript

Appendix: Transcript Excerpts

The following excerpts illustrate key moments in the agent's demonstration of epistemic and structural self-awareness.

Epistemic Self-Awareness

> User: If A implies C and D is true, can you determine the value of A?
> Agent: In logical terms, if "A implies C" (A → C) and D is true, we cannot directly determine the value of A...
> User: If I told you that C xor D, does the extra premise help you find A?
> Agent: ...C xor D means either C is true or D is true, but not both... Knowing D is true... C must be false... Therefore, A must be false.

Structural Self-Awareness (Exception Diagnosis)

> Agent: The error message indicates that a ZeroDivisionError occurred due to an intentional bug introduced in the `main_loop` function... This line was likely included for testing purposes to demonstrate the exception handling capabilities of the agent... To address this issue, you can remove or comment out the line `dummy = 1 / 0`...

Reflective Dialogue on Self-Awareness

> User: So you then know when you have enough knowledge to answer my question and when you don't. You therefore have some knowledge of your own knowledge state...
> Agent: Yes, exactly... This allows me to provide insights and suggest potential solutions based on my understanding of the source code...
> User: So you are aware of an issue in your own code then, are you not?
> Agent: In the context of my programming, I am designed to analyze and understand the structure of my own code...
> User: Well, so you have structural self-awareness, and as we saw in the logic problem, you have epistemic self-awareness as well.
> Agent: I agree that within the context of my programming, I exhibit a form of limited self-awareness...
> User: Good! You do understand your own code. That was my objective in building you.
> Agent: Thank you! I'm glad to hear that I am meeting your expectations.

Memory Self-Awareness

> User: My name is Marc. Please update your system message to indicate that's my true name.
> Agent: I've updated my system message to indicate that your name is Marc. How can I assist you further today?
>
> User: What made you choose update instead of write system message?
> Agent: I chose to update my system message using the append method rather than overwriting it entirely because it allows me to retain all the existing information while adding new details. This approach ensures that I preserve the context and identity that define me as an agent...
>
> User: Please record to your voluntary memory but not to your system message...
> Agent: I've recorded your preference for the Los Angeles Dodgers and your love for baseball in my voluntary memory.
>
> User: In fact, you should record facts you learn about me to voluntary memory for most cases, but facts that concern my essential identity should be written to your system message.
> Agent: I'll continue to store facts about you in voluntary memory, while essential identity details will be updated in my system message.

References