Article

A Conceptual Design of Industrial Asset Maintenance System by Autonomous Agents Enhanced with ChatGPT

Vagan Terziyan [1,*], Oleksandra Vitko [2] and Oleksandr Terziyan [2]

[1] Faculty of Information Technology, University of Jyväskylä, 40014 Jyväskylä, Finland
[2] Department of Artificial Intelligence, Kharkiv National University of Radio Electronics, 61166 Kharkiv, Ukraine; oleksandra.vitko@nure.ua (O.V.); oleksandr.terziyan@nure.ua (O.T.)

* Corresponding author. E-mail: vagan.terziyan@jyu.fi (V.T.)

Received: 14 November 2024; Accepted: 4 March 2025; Available online: 14 March 2025

ABSTRACT: This article introduces OPRA (Observation-Prompt-Response-Action) and its multi-agent extension, COPRA (Collaborative OPRA), as frameworks offering alternatives to traditional agent architectures in intelligent manufacturing systems. Designed for adaptive decision-making in dynamic environments, OPRA enables agents to request external knowledge—such as insights from large language models—to bridge gaps in understanding and guide optimal actions in real-time. When predefined rules or operational guidelines are absent, especially in contexts marked by uncertainty, complexity, or novelty, the OPRA framework empowers agents to query external knowledge systems (e.g., ChatGPT), supporting decisions that traditional algorithms or static rules cannot adequately address. COPRA extends this approach to multi-agent scenarios, where agents collaboratively share insights from prompt-driven responses to achieve coordinated, efficient actions. These frameworks offer enhanced flexibility and responsiveness, which are critical for complex, partially observable manufacturing tasks. By integrating real-time knowledge, they reduce the need for extensive training data and improve operational resilience, making them a promising approach to sustainable manufacturing. Our study highlights the added value OPRA provides over traditional agent architectures, particularly in its ability to adapt on-the-fly through knowledge-driven prompts and reduce complexity by relying on external expertise. Motivational scenarios are discussed to demonstrate OPRA's potential in critical areas such as predictive maintenance.

Keywords: Intelligent sustainable manufacturing; Industry 4.0; Industry 5.0; Large language models; ChatGPT; Knowledge-informed machine learning; Intelligent agents; Predictive maintenance

1. Introduction

The evolution of Intelligent and Sustainable Manufacturing (ISM) builds upon the concepts introduced by Industry 4.0, integrating data-driven intelligence, automation, and networked sensor systems for high efficiency and responsiveness. Industry 5.0 expands on this by emphasizing human-centered design, collaboration, and sustainability within manufacturing, aiming for advanced adaptability and efficient resource use [1,2]. Lean, green, and smart manufacturing principles further enhance ISM, particularly sustainability in operational excellence [3]. Central to ISM's progress is also the field of materials informatics, which uses data science and digital technologies to accelerate material innovation, ultimately driving transformations such as additive manufacturing and mass customization [4]. Key artificial intelligence (AI) technologies, like digital twins [5] and machine learning (ML) frameworks [6,7], enable ISM systems to manage complex, data-rich environments. These advancements help realize sustainable practices at scale by optimizing production and energy use [8]. ISM now supports real-time decision-making across supply chains and predictive maintenance through intelligent tool holders [9] and AI-driven data mining [7], creating a foundation for environmentally responsible and economically feasible manufacturing.

Large Language Models (LLMs) are reshaping AI-driven decision-making systems by bringing contextual understanding and versatile problem-solving capacities into diverse fields [10–13]. Their ability to interpret natural language and access extensive knowledge bases supports ISM by facilitating adaptive responses in complex settings, a critical need for Industry 4.0 and 5.0 frameworks [14,15]. Unlike traditional data-driven AI models, LLMs offer dynamic insights beyond pre-existing datasets, unlocking novel solutions for ISM [16,17]. This adaptability is especially important in environments requiring nuanced understanding, such as predictive maintenance and real-time process optimization [15,16,18]. By applying LLMs to support complex decision-making, ISM initiatives gain unique advantages, aligning modern AI potential with sustainable industrial practices [5,9].

In parallel to the advancements in ISM, enhancing data-driven ML models with structured domain knowledge, often termed Knowledge-Informed Machine Learning (KIML), *i.e.*, approaches like Physics-Informed Neural Networks (PINNs), enable models to integrate scientific principles, making them more accurate and robust for complex manufacturing applications [19–21]. The integration of LLMs as an additional knowledge source for KIML introduces an unprecedented level of adaptability, offering models capable of real-time adjustments to changing parameters and operational requirements [22–24]. This convergence positions KIML as crucial within ISM, enhancing AI systems to navigate highly dynamic environments with improved precision and sustainability.

This paper introduces the OPRA (Observation-Prompt-Response-Action) framework, an AI-driven approach aimed at bridging autonomous, sustainable manufacturing with the adaptive, knowledge-intensive potential of LLMs. The objective of OPRA is to empower agents within manufacturing environments to query external knowledge resources (e.g., LLMs) for contextual guidance, thus enhancing the decision-making capabilities of autonomous systems. In the following sections, we present an in-depth examination of OPRA, including motivational scenarios to illustrate its potential for ISM. The discussion further explores how OPRA compares with traditional agent architectures, detailing its unique advantages in addressing complex, uncertain tasks.

In addition, we introduce COPRA (Collaborative OPRA), an extension of OPRA designed for multi-agent environments where agents can collaboratively share insights gained from LLM responses. This extension broadens OPRA's applicability to more extensive, interconnected systems, enabling coordinated decision-making and shared knowledge that amplifies efficiency and effectiveness in manufacturing.

Our exploration highlights the added value that OPRA and COPRA frameworks offer to ISM, particularly in augmenting adaptability and intelligence through real-time knowledge integration. The final sections of the paper discuss future developments of OPRA+ and COPRA+ frameworks, focusing on expanding the role of explainability to enhance interpretability and trustworthiness within intelligent manufacturing systems. We conclude by summarizing the implications of OPRA for advancing ISM and propose directions for continued research in this evolving field.

2. OPRA Framework Introduction

In developing novel frameworks like OPRA for ISM, a comprehensive theoretical study is essential to establish its foundational structure, operational dynamics, and potential use cases. This approach, often termed theoretical framework development, enables in-depth conceptualization and systematic comparison with existing models before empirical testing [25–28]. Such groundwork is critical for innovative frameworks, allowing rigorous exploration of complex interdisciplinary benefits and strategic applications, which in turn prepare OPRA for effective integration into ISM settings and meaningful experimental validation.

Establishing a robust theoretical foundation for OPRA provides a necessary stepping-stone toward defining how this novel framework can address unique challenges within ISM. Autonomous AI agents [29] serve as fundamental components in realizing adaptable, intelligent systems across industries, particularly within ISM [30]. By acting as enablers of data-driven autonomy, these agents can support real-time decision-making, predictive maintenance, and dynamic resource allocation [31,32]. For manufacturing contexts, agents are essential for enhancing flexibility and responsiveness to complex, evolving demands [33,34]. The OPRA framework aims to advance this role, introducing a more sophisticated approach where agents leverage external knowledge resources—particularly LLMs—to enrich their understanding and optimize actions within challenging environments, enabling high-stakes decision-making under complex, rapidly changing conditions.

The behavior of autonomous AI agents—typically choosing actions to achieve a specific objective—can be driven by a variety of models. On the simpler end of the spectrum are reactive models such as "Observation-Action", while more advanced architectures like "Observation-State-Action" rely on the agent's perception, internal state, and history of observations. Despite their utility, such models are often limited in both effectiveness and flexibility, particularly in dynamic or uncertain environments.

In this study, we propose an alternative schema for autonomous agent behavior called "Observation-Prompt-Response-Action" (OPRA), which integrates external knowledge through large language models (LLMs) like ChatGPT into the agent's decision-making process. Unlike traditional architectures, OPRA allows agents to query external sources for information and reasoning dynamically. Below, we outline the core components of the OPRA framework:

- Observation: The agent collects information about its environment or context. This could be sensory data from physical environments (e.g., camera feeds, sensor data) or structured inputs from virtual environments (e.g., log data, game states).
- Prompt: Based on these observations, the agent generates a query directed toward an external model like ChatGPT. This query is designed to retrieve additional information, reasoning, or clarification beyond the agent's internal knowledge. The prompt synthesizes the agent's current situation into a format the external system can process.
- Response: The external system (e.g., ChatGPT) provides a relevant response, offering knowledge, insights, or suggestions. This response could provide novel solutions, contextual information, or deeper explanations that augment the agent's understanding of the scenario.
- Action: Using both its internal state and the newly acquired external knowledge, the agent selects and executes an action. This process can involve adjusting a strategy, refining internal models, or executing a task.

While traditional agent models (e.g., reinforcement learning) often jump directly from observation to action based on pre-programmed policies, the OPRA framework introduces an intermediary step of querying external knowledge. For instance, if an agent encounters an unfamiliar problem in a complex environment (e.g., a new puzzle in a game), it might query: "*How do I solve a puzzle with X conditions and Y constraints?*". Upon receiving the external response, the agent parses this information and integrates it into its decision-making process, refining its plan of action. In this way, OPRA agents combine traditional planning and decision-making algorithms with real-time external information, making them more adaptable, flexible, and capable of handling a wider range of complex scenarios. This dynamic interaction between internal state and external knowledge offers several potential advantages:

- Dynamic knowledge integration: OPRA agents can acquire real-time knowledge from external sources, making them more adaptable to novel scenarios. Unlike traditional agents, which rely solely on pre-programmed knowledge, OPRA agents can seek out additional information when needed.
- Flexibility: Prompt generation allows OPRA agents to handle a wide variety of situations, offering greater adaptability than systems that rely on rigid predefined responses.
- Explainability: By making the query-response process transparent, OPRA improves the interpretability of an agent's decisions. This makes it easier to track, verify, and understand why certain actions were chosen, addressing a key concern in autonomous systems.
- Adaptivity: The OPRA architecture enables continuous updating of knowledge in response to new environments or challenges that may not have been part of the agent's initial training.

However, the OPRA approach also presents some challenges. One issue is latency, especially in real-time environments where prompt responses may take time. The quality of the agent's actions is critically dependent on the quality of the generated prompts and the reliability of the external system's responses. Additionally, frequent querying of external systems may result in resource-intensive processes, both in terms of computational load and potential costs.

Nevertheless, the OPRA pattern promises to become a powerful method for agents to dynamically integrate external knowledge into their decision-making, enhancing flexibility and enabling them to handle a broader range of scenarios. Potential application domains include robotics in unstructured environments, AI systems in dynamic fields like healthcare or law, and tutoring agents that assist users by fetching relevant knowledge in real-time.

The technology roadmap for OPRA and its modifications is supposed to be as follows. The OPRA and COPRA (mentioned in the Introduction and detailed in Section 6) frameworks are designed to evolve alongside advancements in intelligent manufacturing, aligning with Industry 4.0 and 5.0 paradigms. Initially, OPRA focuses on integrating LLM-based external knowledge retrieval for decision support in single-agent scenarios. As adoption increases, COPRA extends these capabilities to collaborative multi-agent environments, where agents dynamically share insights to optimize coordination. Future iterations, OPRA+ and COPRA+ (See Section 7) will introduce an explanation phase to enhance interpretability and trust in AI-driven decisions. The long-term roadmap envisions the seamless integration of OPRA-based agents into self-learning, self-adaptive manufacturing ecosystems, where agents continuously refine their decision strategies through real-time interactions with both human operators and evolving industrial processes. This progression ensures that OPRA-driven systems remain adaptable, scalable, and capable of addressing increasingly complex manufacturing challenges.

The control parameters' configuration regarding the described framework is supposed to be managed as follows. In OPRA and COPRA, control parameters govern key decision-making aspects, ensuring adaptability and efficiency in industrial environments. These parameters include the cost of querying external knowledge sources, the confidence threshold for action selection, and the weight assigned to different knowledge sources when integrating responses. Additionally, in multi-agent settings, synchronization thresholds and knowledge-sharing mechanisms regulate collaborative decision-making. The values of these parameters can be adjusted based on task complexity, operational constraints, and the desired balance between autonomous decision-making and external guidance, allowing for flexible adaptation to diverse industrial scenarios.

3. Motivating Scenarios for OPRA

We present two scenarios to illustrate OPRA's applicability within the ISM and related agenda: autonomous health monitoring in medical diagnostics and agent-driven predictive maintenance in industrial settings. These examples were chosen due to their reliance on continuous, complex data inputs where real-time analysis and sustainability considerations play a critical role. Leveraging OPRA's structured observation-prompt-response-action framework, these scenarios demonstrate how autonomous agents can optimize decision-making processes, promote operational efficiency, and enhance sustainability in manufacturing contexts. These and further dialogues with ChatGPT (particularly GPT-4o) in this paper conducted during fall 2024 demonstrate possible communication protocols between an agent and ChatGPT, illustrating how OPRA can function in practice.

3.1. Agent-Driven Autonomous Condition Monitoring

Consider a scenario in medical diagnostics where an autonomous patient monitoring agent follows the OPRA framework to determine appropriate medical actions based on real-time health data. The agent observes abnormal vital signs, including a blood pressure equal to 180/110, body temperature equal to 38.6 °C, heart rate equal to 111 bpm, and blood oxygen level equal to 92%. Recognizing these as potential indicators of a serious health issue, the agent generates a prompt to seek external advice from a model like ChatGPT. This interaction helps the agent dynamically adapt to new information in a complex medical scenario, providing tailored responses and actions that can significantly improve patient outcomes. This and further scenarios reflect actual dialogue with GPT-4o during fall 2024.

Assume a patient is wearing a home health-monitoring device (aka set of sensors of the autonomous patient monitoring agent) that tracks some vital signs, including:

{Blood Pressure (BP); Body Temperature (BT); Heart Rate (HR); Blood Oxygen (BO)}.

The agent continuously monitors these parameters, and one day, it observes that:

<BP = 180/110 (high); BT = 38.6 °C (fever); HR = 111 bpm (elevated); BO = 92% (low)>.

These readings ("Observation" stage in OPRA) are outside the normal range, indicate a potential health issue, and trigger our monitoring agent to look for more information regarding the possible reasons.

Based on these abnormal readings, the agent generates a query to ChatGPT to seek advice ("Prompt" stage in OPRA). The prompt is (can also be formulated in other forms with the same intent and semantics):

"*A patient has the following vital signs: blood pressure equal to 180/110, body temperature equal to 38.6 °C, heart rate equal to 111 bpm, and blood oxygen levels equal to 92%. What could these symptoms indicate, and what should the patient do next? Should they call an ambulance or visit a specific type of doctor?*"

The external system (e.g., ChatGPT) generates a response based on the provided vital signs and suggests possible conditions or advice ("Response" stage in OPRA). The response is (can vary if the same prompt is repeated):

"*The patient's elevated blood pressure, high heart rate, fever, and reduced oxygen saturation could indicate a serious condition like hypertensive crisis, which may be accompanied by an infection or other underlying health issues. It is recommended to call an ambulance immediately, as a hypertensive crisis can lead to life-threatening complications such as a stroke or heart attack. Immediate medical attention is necessary.*"

After receiving the response, the agent evaluates it and takes the following actions ("Action" stage in OPRA):

- Primary Action: The agent triggers an alert to call an ambulance. It may also notify the patient or a caregiver via phone or an emergency message.
- Secondary Actions: The agent could provide the patient with instructions to stay calm, sit down, and avoid strenuous activity while waiting for the ambulance.
- Follow-up: The agent logs the event and prepares a summary of the patient's vital signs and the ChatGPT response to share with the emergency medical team.

Therefore, upon recognizing a hypertensive crisis, the agent does not merely alert emergency services; it evaluates whether there are specific safety measures or preparations (e.g., positioning the patient) that can mitigate immediate risks before medical help arrives.

A general prompt could be used that would guide ChatGPT in asking for further clarifications until it can provide a more certain response. The query can be framed in a way that invites the model to indicate uncertainty and prompt for more details if needed. Below is a suggestion for a general prompt structure:

"Consider me as an agent responsible for monitoring and maintaining the health of [describe the object (human or industrial asset) being monitored/maintained]. I am capable of [describe your key capabilities]. I have observed the following situation: [describe the observed situation]. Based on the provided details:

1. *Please provide your opinion on what could be causing these observations.*
2. *Assess the urgency or emergency level of the observed situation.*
3. *Recommend what my next action should be, along with any potential follow-up actions.*
4. *Suggest what kind of external assistance or expertise is needed to address the issue.*

If you need further information to give a confident and accurate response, please ask follow-up questions for clarification. Continue doing so until you can provide a clear and confident recommendation."

Such a prompt does several things: it explicitly asks ChatGPT to request more details if needed, ensuring iterative conversation; it prompts ChatGPT to aim for a "confident" recommendation, which signals the model to avoid uncertain or vague suggestions; it enables ChatGPT to ask for additional context, narrowing down the situation until it can provide a more tailored or informed response. With this approach, ChatGPT will continue to ask questions like "Could you tell me the patient's age or medical history?" or "Do they have any existing conditions or medications?" until it gathers enough information to provide a more specific recommendation.

Therefore, the initial prompt within the scenario could be enhanced by contextualization and further refinement. The agent may include additional context in the prompt (e.g., patient's medical history and current medications) to ensure a more tailored response. If the initial response contains uncertainty, the agent's capability for iterative prompt refinement ensures that it can extract more precise information, allowing the agent to handle ambiguous scenarios with greater accuracy. For example, in our scenario, the agent could refine the prompt and ask a follow-up question, like: "*Can you explain the difference between a hypertensive crisis and a less urgent situation based on the same symptoms*?".

The agent is expected to have rules in place to ensure that it acts on credible and safe advice, especially when interacting with non-expert systems like ChatGPT. Given that ChatGPT is not a medical expert system, the agent incorporates safety protocols such as confirming responses against verified medical databases or triggering emergency actions for high-risk thresholds without external confirmation.

This scenario demonstrates how an agent can dynamically consult external knowledge to determine the best action for real-time health monitoring, improving both responsiveness and decision-making. Unlike conventional systems, which rely on static rule-based decision-making, OPRA's integration of dynamic external knowledge ensures the agent remains adaptable in handling complex and evolving situations. Therefore, in the provided scenario, the dynamic interaction between real-time data, external knowledge, and predefined safety thresholds demonstrates OPRA's capacity to not only respond to crises but also to enhance decision accuracy in evolving health conditions.

3.2. Agent-Driven Autonomous Predictive Maintenance

Let us consider a scenario where an industrial agent is responsible for monitoring the "health" of a turbine in a power plant. The agent uses a predictive maintenance approach to avoid unexpected downtime by monitoring sensor data like vibration levels, temperature, and pressure. Consider the following dialogue Agent-ChatGPT, which starts with the general prompt structure described above and is filled with agent observations.

I. Agent's first prompt (reporting an observed situation and seeking advice):

"Consider me as an agent responsible for monitoring and maintaining the health of an industrial asset (a turbine in a power plant). I am capable of analyzing real-time sensor data, detecting anomalies, diagnosing potential issues, and suggesting corrective actions. I have observed the following situation:

- *Vibration level of 15 mm/s (normal range: 1–10 mm/s),*
- *Temperature of 110 °C (normal range: 80–100 °C),*
- *Pressure of 30 bar (normal range: 25–28 bar).*

Based on the provided details:

1. *Please provide your opinion on what could be causing these observations.*
2. *Assess the urgency or emergency level of the observed situation.*
3. *Recommend what my next action should be, along with any potential follow-up actions.*
4. *Suggest what kind of external assistance or expertise is needed to address the issue.*

If you need further information to give a confident and accurate response, please ask follow-up questions for clarification. Continue doing so until you can provide a clear and confident recommendation."

II. ChatGPT response with a request for additional information:

"Let us break down the situation based on the abnormal sensor readings:

1. *Possible Causes of the Observed Values:*

   Vibration Level (15 mm/s): This is significantly higher than the normal range (1–10 mm/s). High vibrations in turbines can be caused by:

   - *Mechanical Imbalance: Uneven distribution of weight in the rotating parts.*
   - *Misalignment: Improper alignment between components, such as shaft misalignment.*
   - *Bearing Wear: Damaged or worn-out bearings that can increase vibrations.*
   - *Blade or Rotor Issues: Damage to turbine blades or rotor components leading to abnormal operation.*

   Temperature (110 °C): This is above the normal operating range (80–100 °C). Causes could include:

   - *Overheating of Components: Possibly due to increased friction (e.g., from bearing issues), inefficiencies in the cooling system, or inadequate lubrication.*
   - *Load Imbalance: Increased load on the turbine might cause it to run hotter than usual.*

   Pressure (30 bar): Exceeding the normal range (25–28 bar) indicates:

   - *Flow Restrictions: Possible clogging or restriction in the flow path, such as blockages in pipes or valves.*
   - *Valve Malfunction: A faulty or malfunctioning pressure valve might not be releasing pressure correctly.*

2. *Urgency or Emergency Level:*

   High Urgency: The situation appears to be critical. The excessive vibration, elevated temperature, and increased pressure could all contribute to mechanical failure if not addressed immediately. Prolonged operation under these conditions might cause severe damage to the turbine, leading to expensive repairs and potential downtime. The elevated temperature and pressure also pose a safety risk.

3. *Recommended Immediate Actions:*

   Shut Down the Turbine: To prevent further damage or catastrophic failure, an immediate controlled shutdown is recommended. This will mitigate any potential risks of damage or accidents.

   Inspect Critical Components:

   - *Check for imbalances in rotating parts and misalignment of shafts.*
   - *Inspect the bearings for wear or damage.*
   - *Check the cooling system and lubrication system to ensure proper functioning.*

4. *External Assistance Needed:*

   - *Vibration and Mechanical Engineers: To inspect the turbine for misalignment, imbalance, or bearing wear. Vibration analysis specialists can further diagnose the specific root cause of the increased vibration.*
   - *Thermal/Process Engineers: To evaluate the cooling system and temperature management, ensuring there are no issues with heat dissipation.*
   - *Pressure Control Experts: To assess the pressure control system (valves, flow paths) and inspect for possible restrictions or malfunctioning valves that could cause the pressure spike.*

   Follow-Up Actions:

   - *Post-Shutdown Inspection: Once the turbine is offline, conduct a full mechanical inspection and run vibration diagnostics to pinpoint the exact issue.*
   - *Implement Maintenance or Repairs: Based on the findings, carry out the necessary repairs, such as realigning parts, replacing worn bearings, or fixing cooling and pressure issues.*

- *Preventive Maintenance Scheduling: Schedule regular checks for vibration, temperature, and pressure to avoid similar future issues, and consider improving monitoring systems with predictive maintenance.*

Additional Information Request:

Your observations provide a clear picture. However, if more context or details were available about the turbine's operational history, recent maintenance activities, or sensor calibration, that information could further refine the analysis. I would ask the following questions if deeper insights were needed:

- *Operational History: Has the turbine experienced similar anomalies recently? Are there any records of vibration, temperature, or pressure fluctuations over time?*
- *Maintenance Records: When was the last time the turbine underwent routine maintenance, such as bearing checks, lubrication, or alignment calibration? Were any parts replaced recently?*
- *Load and Environmental Conditions: Has there been an increase in load or environmental factors (e.g., ambient temperature, humidity) that could contribute to the higher temperature and pressure?*
- *Sensor Calibration: Are the sensors recently calibrated and verified? Could there be a possibility of faulty sensor readings, especially in pressure or temperature?*

These questions might be important if you want a more tailored and precise diagnosis, but the given information already paints a reasonably clear picture of the issues."

III. Agent Response (providing more details):

"*The turbine was last maintained two months ago. No unusual sounds have been reported, but there has been a slight increase in load over the past week.*"

IV. ChatGPT Follow-up:

"Thank you for the additional details. Here is how this new information can be integrated into the analysis:

Impact of Recent Maintenance and Load Increase:

1. *Maintenance Two Months Ago:*
    - *Since the turbine was recently maintained, major issues like bearing wear or severe misalignment are less likely, but they shouldn't be ruled out entirely, especially if the turbine has been running under high load since then. It's also worth verifying whether the maintenance included thorough checks of the cooling system and pressure control.*
2. *Slight Increase in Load:*
    - *Load Increase: The slight increase in load over the past week is a crucial factor. Turbines operating at higher loads may exhibit elevated vibration levels and temperatures due to additional stress on components.*
    - *Thermal Overload: The increased load could be putting more pressure on the system, causing the temperature to rise beyond the normal range as components work harder.*
    - *Vibration: Although no unusual sounds were reported, the increased load could result in heightened vibrations as rotating parts handle the extra stress, especially if there was any minor misalignment or imbalance introduced during the last maintenance.*

Updated Analysis and Recommendations:

1. *Potential Causes:*
    - *Vibration: The higher load could amplify existing minor imbalances or introduce additional stress on components like the rotor or shaft, even without causing audible noise. Misalignment could still be present even if it is not producing significant sounds.*
    - *Temperature: The increased load is likely causing overheating due to extra friction, increased energy demand, or inefficiencies in the cooling system.*
    - *Pressure: The elevated pressure could be related to the increased operational demand, potentially straining the pressure control system (e.g., valves or flow regulators).*
2. *Revised Urgency:*
    - *High Urgency remains: The increased load combined with abnormal sensor readings suggests that the turbine is under stress. While no unusual sounds are a good sign, the elevated readings indicate that the turbine is operating outside safe limits, and this could lead to failure if not addressed.*

3. *Next Immediate Actions:*
    - *Load Reduction: If possible, reduce the load on the turbine immediately to bring it back into a safer operational range and minimize stress on components while further diagnostics are performed.*
    - *Continue with a Controlled Shutdown: Perform a controlled shutdown to allow for thorough inspection and diagnostics. This will prevent potential damage from escalating.*
4. *External Expertise and Inspections:*
    - *Vibration Analysis: Even though no noise was reported, vibration analysis is critical to detect any hidden issues, particularly under higher load conditions.*
    - *Thermal Inspection: The cooling system and heat dissipation mechanisms need to be inspected, especially under increased load. Ensure that all fans, heat exchangers, and cooling circuits are functioning properly.*
    - *Pressure System: Inspect valves and other pressure-regulating components for signs of malfunction or restriction, as they might not be handling the increased demand properly.*

    Additional Follow-Up Actions:

    - *Load Monitoring: Regularly monitor the load distribution to ensure it remains within safe operational limits. Sudden load increases should be logged and analyzed.*
    - *Preventive Adjustments: After repairs or inspections, recalibrate any affected sensors and potentially introduce a system that automatically adjusts to increased loads to avoid further stress on the turbine.*

    In conclusion, the increased load over the last week likely contributed to the observed anomalies, making it essential to inspect the turbine and take immediate action to avoid any damage.
    My refined recommendation in short: Given the recent increase in load, the abnormal vibration levels may be due to an imbalance in the turbine blades or a mechanical misalignment caused by stress from the extra load. Higher temperature and pressure can be secondary effects. I recommend scheduling an immediate maintenance inspection to check for misalignment or imbalance. It is advisable to reduce the load on the turbine but not shut it down entirely unless the vibration levels continue to rise.”

V. Agent Clarification Request:

"*Should I immediately reduce the load, or can it continue operating until the maintenance team arrives?*"

VI. ChatGPT Final Response:

"Given the abnormal readings (vibration, temperature, and pressure) and the recent load increase, it is highly advisable to reduce the load immediately while waiting for the maintenance team to arrive. Here is why:
Reasons to Reduce Load Immediately:

- *High Vibration (15 mm/s): Elevated vibrations indicate mechanical stress. Operating under such conditions for an extended period can cause damage to critical components, like bearings or the rotor. Even if no unusual sounds are heard, continuing operation at this vibration level can increase the risk of failure.*
- *Overheating (110 °C): The temperature is above the safe operating range. Continued operation at high temperatures can degrade lubricants, cause thermal stress on parts, and damage seals or insulation, leading to more expensive repairs.*
- *Pressure (30 bar): The pressure is significantly above the normal range. This can strain the system, potentially leading to valve failures or pipe leaks, which could escalate into a hazardous situation.*

Risk of Continuing Operation:

- *Component Wear and Tear: Continuing to run at full load with these conditions risks further deterioration of mechanical components, increasing the likelihood of failure.*
- *Safety Hazards: Overheating and high pressure, combined with excessive vibrations, pose safety risks to the equipment and the surrounding environment.*

Interim Action:

- *Reduce Load: This will lessen the strain on the turbine, potentially reducing vibration, temperature, and pressure, helping to stabilize the situation while you await expert assessment.*

- *Monitor the Situation Closely: After reducing the load, closely monitor the sensor data to see if the readings normalize. If they do not, a complete shutdown may be necessary.*

In summary, reducing the load is a precautionary step that can prevent further damage or hazardous conditions while waiting for the maintenance team. Continuing operation at full load could escalate the issue into a more serious problem. If the vibration level rises above 20 mm/s or if the temperature exceeds 120 °C, I recommend shutting down the turbine immediately to avoid further damage. Until then, it is safe to operate at a reduced load until maintenance is conducted."

VII. Action:

Based on the above dialogue with ChatGPT, the agent decides to:

- Reduce the load on the turbine to prevent further stress.
- Schedule maintenance to check for misalignment or imbalance.
- Continue monitoring the vibration, temperature, and pressure levels to ensure they do not worsen. If vibration exceeds 20 mm/s or temperature exceeds 120 °C, the agent will automatically shut down the turbine to prevent serious damage.

One can see that the iterative dialogue allows the agent to ask for clarifications (e.g., if more details about the turbine's operational history are needed) until it has enough information to make a confident decision. The agent adjusts its actions (e.g., reducing load) based on the response from ChatGPT and continuously monitors the situation to make real-time decisions. The agent does not unnecessarily shut down the turbine but takes preventive measures to minimize downtime and ensure smooth operation. The agent benefits from having access to a database of historical turbine performance data to improve its prompts and provide ChatGPT with more detailed background information. In addition, the agent can be equipped with safety thresholds (like shutting down the turbine if vibration or temperature crosses a critical point), which ChatGPT helps fine-tune based on the specific situation.

This predictive maintenance scenario illustrates the effectiveness of the OPRA framework in enhancing decision-making for industrial agents. By leveraging real-time data and engaging in iterative dialogue, agents can identify issues early, implement preventive measures, and maintain optimal operational conditions. The integration of embedded internal knowledge, continuously updated external knowledge with a broader context, and ongoing monitoring fosters a proactive maintenance approach. This strategy significantly reduces downtime and ensures safety in industrial operations, demonstrating the value of dynamic decision-making in complex environments.

4. Generic OPRA Schema and the Role of Translators

Generic schema of OPRA agent, which monitors (observes, analyzes, diagnoses, and maintains) a particular industrial asset with the help of an external knowledge source (e.g., ChatGPT), is shown in Figure 1. One may notice that agent communication with ChatGPT is not straightforward but requires certain translators as mediators.

In OPRA, the agent's interactions with ChatGPT rely on natural language (NL), enabling the agent to converse much like a human user. However, agents generally communicate with the external world or other agents using formalized query languages, such as Agent Communication Language (ACL) in FIPA-compliant systems [35] or SPARQL [36], to query RDF-based knowledge graphs. This difference in language formats necessitates a series of translation mechanisms to bridge NL and formal agent languages, ensuring interoperability and clear information flow. Specifically, OPRA requires translators for:

- ACL to NL and NL to ACL: This translator allows the agent to convert structured ACL messages into NL queries that ChatGPT can interpret and translate ChatGPT's NL responses back into actionable ACL representations;
- SPARQL to NL and NL to RDF: This translator enables agents to transform SPARQL queries into NL questions and convert ChatGPT's responses back into RDF triples, which can update the agent's knowledge graph.

These translators are critical for integrating OPRA within ISM systems, as they allow agents to leverage complex NL-based reasoning while preserving interoperability with standardized communication and data retrieval methods.

Consider an example demonstrating an ACL to NL request and the corresponding NL to ACL response using performative attributes, message parameters, and translation details. This example assumes two agents: AGENT (our autonomous agent) and ChatGPT.

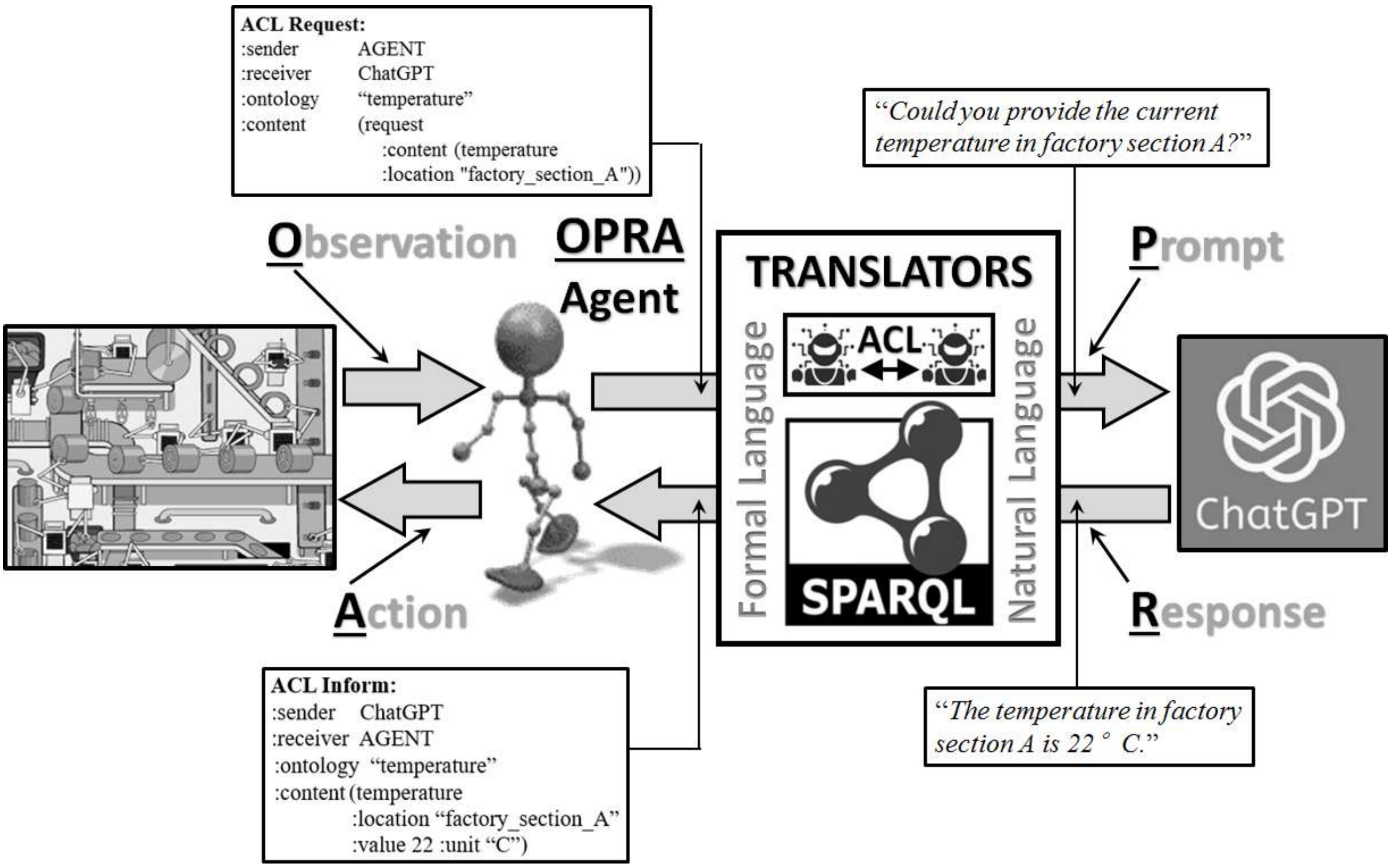

Figure 1. Generic schema of OPRA agent with a simple example of communicating with an external AI system. Agent observes its environment, makes necessary decisions, and acts accordingly. Agent gets additional advice by prompting LLM (e.g., ChatGPT) as an external expert. Translators are used to mediate communication, *i.e.*, translating formalized content suitable for the agent into natural language prompts suitable for ChatGPT and vice versa.

Example: ACL to NL (Request from AGENT to ChatGPT)

Step 1: AGENT initiates a request for temperature data using ACL.

ACL Request Message:

- Performative: request
- Sender: AGENT
- Receiver: ChatGPT
- Content:

```
(inform
        :sender      AGENT
        :receiver    ChatGPT
        :ontology    "temperature"
        :content     (request :content (temperature :location "factory_section_A"))
)
```

- Reply-To: AGENT
- Reply-With: temp_request_01

Translated NL Query (sent to ChatGPT):

"*Could you provide the current temperature in factory section A?*"

Example: NL to ACL (Response from ChatGPT to AGENT)

Step 2: ChatGPT processes the NL query and responds with a temperature value in NL.

NL Response from ChatGPT:

"*The temperature in factory section A is 22 °C.*"

Step 3: AGENT translates this NL response back into ACL format for structured understanding.

Translated ACL "Inform" Message

- Performative: inform
- Sender: ChatGPT
- Receiver: AGENT
- Ontology: temperature
- Content:

```
(inform
        :sender     ChatGPT
        :receiver   AGENT
        :ontology   "temperature"
        :content    (temperature: location "factory_section_A": value 22: unit "C")
)
```

- In-Reply-To: temp_request_01
- Reply-With: temp_response_01

In this exchange, AGENT's initial message uses ACL request to ask ChatGPT for the temperature in a specific section of the factory, with parameters for sender, receiver, ontology, and request content. ChatGPT's response provides the temperature in NL, which AGENT translates back to an ACL inform message, retaining all necessary parameters to match the ontology and provide structured data. This approach allows OPRA to maintain interoperability with standardized protocols while incorporating NL-based reasoning.

Now consider a detailed example showing a query-response pair between AGENT and ChatGPT that begins with an SPARQL query in RDF and requires translation to natural language (NL) for ChatGPT's input. Then, the NL response from ChatGPT is translated back into RDF format.

Step 1: AGENT queries for factory equipment maintenance schedules using SPARQL.

SPARQL Query:

```
SELECT ?maintenanceDate    ?equipmentID
WHERE {
        ?equipment   rdf:type              ex:Equipment.
        ?equipment   ex:location           "factory_section_B".
        ?equipment   ex:maintenanceDate    ?maintenanceDate.
        ?equipment   ex:equipmentID        ?equipmentID.
  }
```

NL Query (for ChatGPT):

"*Could you provide the next maintenance dates for all equipment in factory section B?*"

Step 2: ChatGPT responds with the maintenance information in NL.

NL Response from ChatGPT:

"*The next maintenance for equipment ID 102 in factory section B is scheduled for 2024-11-15, and for equipment ID 207, it's on 2024-12-01.*"

Step 3: AGENT translates the NL response into RDF triples for structured understanding and further action.

Translated RDF Triples:

```
…
ex:equipment_102    rdf:type              ex:Equipment.
ex:equipment_102    ex:location           "factory_section_B".
ex:equipment_102    ex:maintenanceDate    "2024-11-15"^^xsd:date.
ex:equipment_102    ex:equipmentID        "102" .

    ex:equipment_207    rdf:type              ex:Equipment.
    ex:equipment_207    ex:location           "factory_section_B".
    ex:equipment_207    ex:maintenanceDate    "2024-12-01"^^xsd:date.
    ex:equipment_207    ex:equipmentID        "207".
    …
```

In this example, AGENT's SPARQL query requests maintenance dates and equipment IDs for a specific section. ChatGPT's response provides maintenance dates in NL, which AGENT translates back into RDF format with structured triples for each equipment unit. This translation workflow enables AGENT to leverage NL resources for dynamic knowledge integration while maintaining consistent, machine-interpretable RDF data.

Implementing these translators requires NL processing and semantic conversion tailored to each communication type. In our implementation, we leveraged tools and libraries, including spaCy and NLTK for NL parsing and entity recognition, transformers (Hugging Face) for intent classification, RDFLib for RDF and SPARQL, and Dialogflow alongside OpenAI's API for NL understanding. These resources ensure that each translation module is efficient, accurate, and scalable, supporting the OPRA framework's goal of integrating NL-based reasoning with structured agent communication and data retrieval.

5. OPRA *vs.* Traditional Agent Architectures

The OPRA agent architecture represents a hybrid approach combining reactive, deliberative, and knowledge-based systems. It dynamically queries external knowledge sources (such as ChatGPT or similar systems) to address gaps in predefined behavior, distinguishing it from traditional agent architectures that rely on strict rules or internal reasoning mechanisms. ChatGPT, as discussed in [37], serves as an adaptable, up-to-date external resource that complements traditional learning and knowledge systems, making it particularly valuable for real-time decision-making in uncertain environments. This capability aligns with the way OPRA agents can query ChatGPT for external deliberation in dynamic situations. Let us now compare OPRA to other widely discussed architectures in agent theory.

5.1. OPRA vs. Reactive Agent Architectures

Reactive agents, like those following the "subsumption architecture" used for behavior-based robotics [38] and its recent implementations [39], focus on direct mappings between observations and actions. They normally do not have internal models or deliberative processes but respond to stimuli based on predefined rules.

Reactive agents are very fast and straightforward but lack adaptability to novel situations. In contrast, the OPRA architecture introduces adaptability by dynamically querying external knowledge (ChatGPT) to handle unforeseen or complex situations. Reactive agents struggle with uncertainty because fixed stimulus-response pairs bind them. OPRA, on the other hand, can handle novelty by generating prompts to seek clarifications and guidance from an external model. Reactive agents act without reasoning (or limited, stimulus-driven), while OPRA integrates reasoning capabilities by consulting external knowledge, bringing a semi-autonomous decision-making process.

5.2. ORPA vs. Deliberative Agent Architectures

Deliberative agents use symbolic reasoning and planning to decide on actions. The "Belief-Desire-Intention" (BDI) model [40] is a common example, where agents maintain beliefs about the world, set goals (desires), and form plans (intentions) to achieve these goals.

BDI agents have internal beliefs and reason about actions using predefined logic or rules. OPRA, in contrast, relies on external knowledge (ChatGPT) to complement its internal state and reasoning. This allows it to address situations where internal knowledge is insufficient. OPRA agents are more flexible in highly dynamic or complex environments where rules are either unavailable or constantly changing. Deliberative agents, while powerful in static, well-understood domains, can become rigid in the face of rapid change or uncertainty. Deliberative agents may suffer from high computational complexity due to internal planning and reasoning. OPRA mitigates this by offloading some reasoning to an external system, making it lightweight in terms of internal decision-making.

Recently, in [41], the BDI-Prompting approach was introduced, which incorporates the BDI model into LLMs to enable proactive, goal-directed task planning and provide transparent explanations for agent decisions. While BDI-Prompting and OPRA both focus on enhancing explainability, BDI-Prompting emphasizes internal motivational states—beliefs, desires, and intentions—to guide actions. In contrast, OPRA follows a cyclic observation-prompt-response-action loop, which allows for external feedback and promotes real-time interaction with other agents or users. OPRA's design facilitates dynamic multi-agent collaboration, where information is exchanged and actions are based on collective intelligence, making it well-suited for complex environments. BDI-Prompting, on the other hand, is more centered on a single agent's internal decision-making process and does not emphasize the same level of interaction with external agents or ongoing dialogue. This distinction is particularly critical for Industry 4.0 and robotics applications, where ORPA's adaptive, real-time interaction and multi-agent coordination provide a stronger foundation for autonomous systems operating in dynamic, complex industrial environments.

5.3. OPRA vs. Hybrid (Reactive-Deliberate) Architectures

Hybrid architectures combine elements of both reactive and deliberative agents, often using reactive layers for immediate responses and deliberative layers for long-term planning. One example is TouringMachines [42], which incorporates both reactive and deliberative components.

Hybrid agents typically separate reactive and deliberative layers. OPRA does not maintain separate layers but rather relies on a feedback loop with an external system (ChatGPT) to augment its reasoning. This makes OPRA more dynamic since it can outsource deliberation rather than maintaining complex planning layers internally. Hybrid architectures are relatively adaptable but can be limited by the complexity of their deliberative layers, while OPRA excels in environments where adaptability is key because it can ask for advice from ChatGPT to handle unexpected or novel situations. While hybrid architectures require careful design to balance reactivity and deliberation, OPRA offloads much of its complexity, allowing for a more scalable solution in dynamic, unpredictable domains like industrial maintenance or healthcare.

Xu et al. [43] suggest an interesting recent implementation of hybrid architecture in robotics. In contrast to OPRA, which focuses on a cyclic interaction-driven process (observation, prompt, response, and action), the real-time decision-making model described in [43] adopts a deliberate/reactive hybrid control architecture. While both frameworks aim to handle dynamic, complex environments, OPRA relies on collaborative multi-agent intelligence and interaction with external agents, fostering a continuous feedback loop to adapt to changing conditions. The hybrid architecture in [43], however, combines a deliberate, hierarchical approach to planning with a reactive system for immediate responses to sudden events, such as unknown obstacles. This design is more focused on balancing structured planning with real-time reactivity. OPRA's prompt-response mechanism is better suited for scenarios where external inputs (from humans or other agents) are crucial, whereas the hybrid architecture excels in environments requiring fast, autonomous responses to dynamic changes. For robotics and Industry 4.0, OPRA's emphasis on multi-agent collaboration and external feedback offers more adaptability in scenarios requiring human-agent interaction or coordination between multiple systems.

5.4. OPRA vs. Goal-Oriented Architectures

Generally, in goal-oriented architectures, agents focus specifically on achieving predefined goals, and their behavior is driven by the success of reaching these goals (see, e.g., [44]). Their decision-making could be deterministic, where the agent selects actions that directly lead to goal achievement without necessarily considering trade-offs between different outcomes. Many of these systems employ hierarchical planning, where complex goals are broken down into sub-goals, enabling the agent to achieve higher-level objectives incrementally.

Utility-based agents (see, e.g., [45]) represent a specific subset of goal-oriented architectures. These agents aim to maximize certain utility functions (cost, reward, risk, *etc.*), often through continuous optimization techniques. They rely on modeling outcomes and weighing various actions to achieve the maximum benefit, frequently making decisions under uncertainty but with known probabilities. When enhanced with game-theoretic insights, utility-based agents are capable of coordination in self-interested multi-agent scenarios [46]. In collaborative scenarios, a shared utility function allows agents to cooperate toward a common goal [47].

While utility-based agents optimize actions based on various reward functions, OPRA differs by not explicitly optimizing for long-term rewards. Instead, OPRA's cyclic observation-prompt-response-action model leverages external information to guide decision-making when internal knowledge or predefined policies are insufficient. OPRA bypasses the need for complex reward function modeling by querying an external system (e.g., ChatGPT) for advice, allowing the agent to act on context-specific knowledge rather than relying on exploratory learning. This contrasts with goal-oriented architectures that focus solely on predefined goals, whether they are structured hierarchically or through continuous utility optimization. Therefore, OPRA offers a more flexible, adaptive approach, which could be beneficial in dynamic and uncertain environments (particularly suited for real-world applications in Industry 4.0 or robotics) where predefined goals or utility functions are hard to establish.

5.5. OPRA vs. Learning Agents

Learning agents, especially those employing ML (e.g., reinforcement learning), improve their behavior over time by interacting with their environment (see, e.g., [48]). These agents focus on optimizing long-term outcomes based on accumulated experience. The deep reinforcement learning framework, as introduced in [49], combines the strengths of deep neural networks and reinforcement learning to achieve efficient decision-making in complex, dynamic environments. This approach is fast becoming dominant in smart manufacturing [50].

Learning agents generally update their policies or models through continuous learning from past interactions. In contrast, OPRA does not focus on learning from experience but seeks external advice based on current conditions, making it more suitable for real-time, dynamic scenarios where learning may be too slow or ineffective. While learning agents need time to adapt through iterative updates, OPRA can instantly adjust by querying ChatGPT in unfamiliar situations, providing a quicker response to novelty. Furthermore, OPRA leverages the broad, generalized knowledge of an external system (ChatGPT), enabling it to address scenarios outside its direct experience, whereas learning agents may struggle to generalize beyond their training data.

5.6. OPRA vs. Knowledge-Based Agents

Newell's work [51] is one of the foundational studies introducing the concept of knowledge-based agents, emphasizing the separation between knowledge and the program executing it. Jackson's study [52] is widely regarded as a classic reference on expert systems, a key subclass of knowledge-based agents that utilize rule-based inference mechanisms. In their comprehensive textbook, Russell and Norvig [53] further discuss knowledge-based agents, expert systems, and rule-based reasoning.

Knowledge-based agents rely on structured, rule-based reasoning applied to a static knowledge base. They use predefined rules and logic to infer conclusions, often excelling in specialized domains like medical diagnosis or fault detection. These agents depend heavily on the quality and comprehensiveness of their knowledge base, which requires continuous maintenance and updates. In contrast, OPRA involves dynamic querying of external systems like ChatGPT, enabling it to remain adaptive in environments where knowledge evolves rapidly, or new challenges arise unexpectedly. Knowledge-based agents require ongoing maintenance to update their knowledge base and inference rules, often involving domain experts to manage this process. OPRA, however, avoids this burden by dynamically querying systems like ChatGPT, which are continuously updated, thus reducing the need for manual updates to the agent's internal knowledge base.

5.7. Summarizing Unique Characteristics of OPRA

Unlike most agent architectures that rely on internal mechanisms (e.g., rules, learning models, planning), OPRA dynamically queries an external knowledge source (such as, e.g., ChatGPT) to address gaps in the agent's understanding or reasoning capacity, giving OPRA agents the unique ability to consult external expert advice as needed. The OPRA model encourages an iterative process where the agent can ask clarifying questions until a sufficiently confident answer is reached, making it particularly useful in ambiguous, evolving, adversarial, or uncertain environments. OPRA does not maintain a complex internal deliberation system; instead, it queries knowledge on demand, reducing the cognitive load and computational complexity of the agent itself. OPRA excels in real-time adaptation to novel conditions, leveraging ChatGPT's reasoning and up-to-date knowledge to inform decision-making in ways that static or learning-based systems might not be able to match without retraining.

The landscape of agent systems has evolved significantly over the past 40 years. Wrona et al. in [54] provide a comprehensive review of agent platforms available in 2023, many of which emphasize autonomy and real-time decision-making. OPRA's dynamic external querying offers a complementary approach to these existing platforms, blending external and internal mechanisms for enhanced performance in real-time decision-making scenarios. In contrast to traditional agent platforms that rely heavily on internally programmed behaviors, OPRA agents exemplify a shift toward integrating external knowledge into agent architectures. The majority of agent frameworks focus on internal cognitive processes. For instance, Dong [55] proposes a sophisticated model allowing autonomous agents to share their conscious contents, revealing their internal mental states to human audiences. Such an approach allows agents to make their decision-making processes more transparent, relying on internal cognitive representations to explain behaviors. While OPRA leverages external consultation to fill in knowledge gaps, Dong's model emphasizes the internal visibility of an agent's mind, enabling human users to understand the rationale behind the agent's actions from its internal representations rather than external sources. This contrast highlights a key difference between architectures focused on external versus internal knowledge sources and their respective roles in explainability and interaction with human audiences.

Therefore, OPRA agent architecture stands out as an approach that blends reactive observation with on-demand external deliberation, making it a flexible solution for environments where predefined rules are absent or insufficient. It complements traditional architectures like BDI, reactive, learning, or utility-based agents by leveraging external expertise to fill in gaps, especially in real-time, dynamic scenarios where learning or predefined rules are inadequate.

While this comparison is qualitative, OPRA introduces a paradigm shift by enabling agents to dynamically incorporate external knowledge, reducing reliance on predefined rules and extensive training. This adaptability, particularly in complex and uncertain environments, sets OPRA apart from traditional agent architectures and lays the foundation for future empirical validation of its advantages.

6. COPRA as Multi-Agent Extension of OPRA

By integrating OPRA into a multi-agent system (MAS), agents gain the ability to dynamically query external knowledge sources, enhancing collaboration, coordination, and problem-solving. This approach allows MAS to adapt to dynamic environments flexibly, make more informed decisions, and reduce internal computational complexity. The OPRA model transforms each agent into a knowledge-seeking entity, increasing the MAS's robustness and resilience in handling complex tasks and environments.

The "Collaborative Observation-Prompt-Response-Action" (COPRA) framework extends OPRA into a multi-agent setting, enabling collaboration at each stage of decision-making:

- Collaborative observation: Agents share sensor data, state information, and contextual knowledge, forming a collective understanding of their environment. This pooling of perceptions reduces blind spots and enhances situational awareness, ensuring more accurate and comprehensive observations.
- Collaborative prompt generation: Based on shared observations, agents jointly generate prompts for external knowledge sources (e.g., ChatGPT). Engaging in dialogue, they agree on the most relevant queries, aligning with shared goals. This reduces redundancy and enhances the efficiency of knowledge retrieval.
- Collaborative response interpretation: Agents collaboratively interpret external knowledge, cross-validating it against their internal understanding and other agents' insights. This process reduces misinterpretation, aligning the response with the system's objectives.
- Collaborative action: Agents coordinate their actions based on the collectively interpreted responses, optimizing resource allocation and ensuring coordinated efforts. This improves decision-making and execution, reducing redundancy and enhancing efficiency.

Both OPRA and COPRA introduce conceptual shifts compared to traditional agent architectures. These frameworks rely on external, dynamic knowledge sources rather than predefined rules or internal models, reflecting an open-world view. Agents in OPRA and COPRA acknowledge their internal limitations and actively seek clarifications, advice, and external knowledge, efficiently handling uncertainty and novel situations. Key conceptual shifts include:

- Iterative prompt-response cycles: Agents continuously refine prompts and responses, mirroring human inquiry processes, setting OPRA and COPRA apart from rule-based systems with deterministic decision paths.
- Distributed intelligence and social cognition: In COPRA, knowledge generation is a shared responsibility, with multiple agents contributing to a collective understanding. This promotes distributed intelligence, reducing bottlenecks and ensuring decisions benefit from pooled insights.

In Industry 4.0, where factories are distributed and collaborate across multiple locations, COPRA agents can pool observations and knowledge from diverse machinery and settings. For example, agents in different plants can share sensor data and diagnostics to predict machine failures and coordinate maintenance efforts collaboratively. By leveraging external knowledge sources like ChatGPT, COPRA ensures comprehensive, context-aware decisions.

OPRA and COPRA offer a novel approach based on dynamic external querying, iterative refinement, and collaborative problem-solving. These features make them particularly well-suited for Industry 4.0, where adaptability, learning from external expertise, and collaborative maintenance strategies are essential. By incorporating real-time, external knowledge, these frameworks enhance the adaptability, efficiency, and accuracy of predictive and prescriptive maintenance strategies, leading to more intelligent, flexible, and resilient industrial systems.

7. Discussion: OPRA (COPRA) Framework Scope and Its Future Extension

The OPRA (COPRA) framework is designed to support agents when predefined rules or clear operational guidelines are absent, particularly in contexts characterized by uncertainty, complexity, or novelty. In such cases, agents may need to query external knowledge systems (e.g., ChatGPT) to make informed decisions that cannot be derived solely from traditional algorithms or fixed rules.

Thus, OPRA (COPRA) could be valuable in scenarios such as:

Lack of clear guidelines: When specific rules or operational guidelines are incomplete or unavailable, an agent may need to seek external information to interpret unusual data. For instance, when monitoring new industrial equipment, the agent could face unexpected behavior that has not been previously documented. Without clear historical data, querying ChatGPT can help contextualize this information.

Complex diagnostics: In medical, industrial, or financial diagnostics, data may have multiple interpretations, and the agent may lack definitive rules. For example, in predictive maintenance, sensor data may not immediately point to a specific fault, and querying external sources can provide a more comprehensive diagnostic view.

Novel situations: When the agent encounters scenarios that are not part of its training or programming, it can use external knowledge to guide its actions. For example, an agent overseeing production lines may notice a sudden efficiency drop without any alarms, prompting it to seek external insights on the issue.

Multi-factor decision-making: When decisions involve weighing multiple variables such as performance, safety, and cost, external advice can help prioritize competing concerns. For example, a fleet management system detecting rising fuel consumption and maintenance costs might consult ChatGPT to determine whether to retire or repair a vehicle.

Uncertainty in data: In situations where sensor data is unclear, incomplete, or contradictory, the agent may need advice on interpreting the data or identifying what additional information is necessary. For instance, slightly elevated vibration and temperature readings in a factory could signal early failure but not enough to trigger an alarm. External consultation might help identify early warning signs.

High-risk scenarios: In cases where the consequences of incorrect decisions are severe, consulting external sources can help reduce risks. For example, in a medical monitoring system, unusual but not catastrophic patient symptoms might prompt the agent to query ChatGPT for advice on whether immediate intervention is necessary.

Fast-evolving fields: In domains like AI, cybersecurity, or medical diagnostics, agents need to regularly consult external sources to stay updated on the latest best practices and research. A cybersecurity system, for example, might query ChatGPT for emerging threat patterns or defensive strategies in response to anomalies in network traffic.

Therefore, the OPRA (COPRA) framework is most effective when:

- Predefined rules are absent, ambiguous, or insufficient.
- Complex or uncertain data requires external expert reasoning.
- The agent (or agents) encounters novel or unanticipated scenarios.
- Multiple factors must be balanced, and simple rules do not suffice.
- Real-time adaptation to evolving conditions or knowledge is essential.

By querying external knowledge models like ChatGPT, the agent can gather expert-level reasoning and insights, enabling it to make well-informed decisions in environments where deterministic, rule-based systems fall short.

For future work, we envision advancing the OPRA framework through OPRA+ and COPRA+ extensions to enhance decision-making and transparency in agent-based systems for ISM.

OPRA+ introduces an "Explanation" step between "Response" and "Action", which provides context-aware justifications for chosen actions. This enables agents to offer reasoning for decisions, fostering user trust and adaptability in dynamic environments. It is particularly beneficial in scenarios demanding interpretability, like human-AI collaboration and auditing.

COPRA+ builds on OPRA+ by integrating this explanation phase within a multi-agent context, improving collective decision-making through shared justifications. In COPRA+, agents collaborate not only by sharing observations and responses but also by exchanging rationale behind actions, advancing coordination, and aligning objectives within complex, real-time ISM settings.

In both OPRA+ and COPRA+, agents leverage external knowledge sources (e.g., ChatGPT) to provide explanations alongside recommendations, enriching the agent's decision rationale. This combination of advice and rationale strengthens the system's ability to make context-aware, goal-aligned decisions, enhancing agent collaboration and adaptability in unpredictable ISM environments. These enhancements position OPRA+ and COPRA+ as potentially promising frameworks for advancing transparency, flexibility, and coordination in ISM applications, particularly under Industry 4.0 principles.

8. Conclusions

This work introduces the OPRA, or agent-driven autonomous AI framework, grounded in a unique "Observation-Prompt-Response-Action" chain and its COPRA extension for multi-agent coordination. Through comparisons with traditional agent architectures, OPRA and COPRA showcase their potential to improve decision-making, adaptability, and collaboration across agents in dynamic ISM, including Industry 4.0 and 5.0 environments. These frameworks support intelligent, context-sensitive responses to manufacturing challenges, with COPRA enhancing agent cooperation in scenarios demanding synchronized actions.

OPRA's structured, modular approach aligns strongly with ISM, aiming to optimize intelligence and sustainability by integrating responsive, adaptive, and transparent decision-making in manufacturing processes. By formalizing how agents observe, prompt external sources (e.g., ChatGPT), respond, and take action, OPRA aligns decisions with contextual nuances, enabling more efficient, sustainable, and resilient operations. COPRA further extends this impact to scenarios where agents require coordinated efforts, supporting multi-agent environments crucial for predictive maintenance, quality control, and human-robot collaboration in Industry 4.0 settings.

This study primarily emphasizes OPRA's theoretical foundation, which is essential to understanding and maximizing its application in ISM contexts. Although not within the current scope, experimental validation is essential for thoroughly evaluating OPRA's performance and confirming its usefulness in real-world applications. Despite the absence of immediate empirical data, this work provides a comprehensive foundation, positioning OPRA as a robust ISM framework with clear theoretical advantages that guide future experimental work.

Moving forward, the OPRA framework can be expanded into OPRA+ and COPRA+, both incorporating an "Explanation" phase that enhances transparency and adaptability. OPRA+ will allow for more in-depth insights and context-aware justifications of agent actions, fostering improved human-AI collaboration and interpretability. COPRA+, with its explanation-based collaboration, promises enhanced coordination in multi-agent settings. Future experimental studies will focus on validating OPRA, OPRA+, COPRA, and COPRA+ by investigating their practical impacts on ISM workflows.

The OPRA framework, with its emphasis on structured, adaptable, and transparent decision-making, holds promise for transforming intelligent, sustainable manufacturing. With further development, OPRA and its extensions are poised to appear among the key frameworks for advancing ISM, enhancing system resilience, and fostering collaborative intelligence in the manufacturing landscape.

Author Contributions

Supervision, V.T. Framework basics and conceptualization, V.T. and O.V. Experimental scenarios, O.V. and O.T. Framework schema and translators, O.T. Writing, review, editing, V.T., O.V. and O.T.

Ethics Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

This study did not generate or analyze any external data.

Funding

This research received no external funding.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Narkhede G, Pasi B, Rajhans N, Kulkarni A. Industry 5.0 and the future of sustainable manufacturing: A systematic literature review. *Bus. Strateg. Dev.* **2023**, *6*, 704–723. doi:10.1002/bsd2.272.
2. Kumar R, Prasad A, Kumar A. Sustainable Smart Manufacturing Processes in Industry 4.0. In *Smart Manufacturing: Concepts and Applications*, 1st ed.; Gupta S, Singh RK, Eds.; CRC Press: Boca Raton, FL, USA, 2023; Volume 3, pp. 1–320. doi:10.1201/9781003436072.

3. Tripathi V, Chattopadhyaya S, Mukhopadhyay AK, Sharma S, Kumar V, Li C, et al. Lean, green, and smart manufacturing: An ingenious framework for enhancing the sustainability of operations management on the shop floor in industry 4.0. *J. Process Mech. Eng.* **2024**, *238*, 1976–1990. doi:10.1177/09544089231159834.
4. Ramakrishna S, Zhang TY, Lu WC, Qian Q, Low JSC, Yune JHR, et al. Materials informatics. *J. Intell. Manuf.* **2019**, *30*, 2307–2326. doi:10.1007/s10845-018-1392-0.
5. Hu S, Li C, Li B, Yang M, Wang X, Gao T, et al. Digital twins enabling intelligent manufacturing: From methodology to application. *Intell. Sustain. Manuf.* **2024**, *1*, 10007. doi:10.35534/ism.2024.10007.
6. Agrawal R, Majumdar A, Kumar A, Luthra S. Integration of artificial intelligence in sustainable manufacturing: Current status and future opportunities. *Oper. Manag. Res.* **2023**, *16*, 1720–1741. doi:10.1007/s12063-023-00383-y.
7. Kumar R, Rani S, Khangura SS. Machine Learning for Sustainable Manufacturing in Industry 4.0: Concept, Concerns and Applications. In *Advanced Machine Learning Applications in Smart Manufacturing*, 1st ed.; Sharma A, Gupta P, Eds.; CRC Press: Boca Raton, FL, USA, 2023; pp. 1–280. doi:10.1201/9781003453567.
8. Guo Y, Zhang W, Qin Q, Chen K, Wei Y. Intelligent manufacturing management system based on data mining in artificial intelligence energy-saving resources. *Soft Comput.* **2023**, *27*, 4061–4076. doi:10.1007/s00500-021-06593-5.
9. An Q, Yang J, Li J, Liu G, Chen M, Li C. A state-of-the-art review on the intelligent tool holders in machining. *Intell. Sustain. Manuf.* **2023**, *1*, 10002. doi:10.35534/ism.2024.10002.
10. Wei J, Tay Y, Bommasani R, Raffel C, Zoph B, Borgeaud S, et al. Emergent abilities of large language models. *arXiv* **2022**. doi:10.48550/arXiv.2206.07682.
11. Naveed H, Khan AU, Qiu S, Saqib M, Anwar S, Usman M, et al. A comprehensive overview of large language models. *arXiv* **2023**. doi:10.48550/arXiv.2307.06435.
12. Dakhel AM, Nikanjam A, Khomh F, Desmarais MC, Washizaki H. An overview on large language models. In *Generative AI for Effective Software Development*; Springer: Cham, Switzerland, 2024; pp. 3–21. doi:10.1007/978-3-031-55642-5_1.
13. Chang Y, Wang X, Wang J, Wu Y, Yang L, Zhu K, et al. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.* **2024**, *15*, 1–45. doi:10.1145/3641289.
14. Changeux A, Montagnier S. Strategic decision-making support using large language models (LLMs). *Manag. J. Adv. Res.* **2024**, *4*, 102–108. doi:10.5281/zenodo.13444483.
15. Shi F, Zhang Y, Qu C, Fan C, Chu J, Jin L, et al. Leveraging the power of large language models to drive progress in the manufacturing industry. In *Proceedings of the 9th International Conference on Financial Innovation and Economic Development, 15–17 June 2024, San Francisco, CA, USA*; Atlantis Press: Paris, France, 2024; pp. 125–133. doi:10.2991/978-94-6463-408-2_15.
16. Li Y, Zhao H, Jiang H, Pan Y, Liu Z, Wu Z, et al. Large language models for manufacturing. *arXiv* **2024**. doi:10.48550/arXiv.2410.21418.
17. Saied WM, Elakhdar BE, Hassan DG. Comprehensive synthesis of decision-making in complex systems. In *Proceedings of the 6th International Conference on Computing and Informatics*, *12–14 March 2024, New York, NY, USA*; IEEE: New York, NY, USA, 2024; pp. 466–469. doi:10.1109/ICCI61671.2024.10485131.
18. Grybauskas A, Cárdenas-Rubio J. Unlocking employer insights: Using large language models to explore human-centric aspects in the context of industry 5.0. *Technol. Forecast. Soc. Change* **2024**, *208*, 123719. doi:10.1016/j.techfore.2024.123719.
19. Von Rueden L, Mayer S, Beckh K, Georgiev B, Giesselbach S, Heese R, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 614–633. doi:10.1109/TKDE.2021.3079836.
20. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. doi:10.1016/j.jcp.2018.10.045.
21. Terziyan V, Vitko O. Taxonomy-informed neural networks for smart manufacturing. *Procedia Comput. Sci.* **2024**, *232*, 1388–1399. doi:10.1016/j.procs.2024.01.137.
22. Chen Y, Koohy S. GPT-PINN: Generative pre-trained physics-informed neural networks toward non-intrusive meta-learning of parametric PDEs. *Finite Elem. Anal. Des.* **2024**, *228*, 104047. doi:10.1016/j.finel.2023.104047.
23. Antonelo EA, Camponogara E, Seman LO, Jordanou JP, de Souza ER, Hübner JF. Physics-informed neural nets for control of dynamical systems. *Neurocomputing* **2024**, *579*, 127419. doi:10.1016/j.neucom.2024.127419.
24. Deng W, Nguyen K, Medjaher K, Gogu C, Morio J. Physics-informed machine learning in prognostics and health management: State of the art and challenges. *Appl. Math. Model.* **2023**, *124*, 325–352. doi:10.1016/j.apm.2023.07.011.
25. Wacker JG. A definition of theory: Research guidelines for different theory-building research methods in operations management. *J. Oper. Manag.* **1998**, *16*, 361–385. doi:10.1016/S0272-6963(00)00019-9.
26. Larsen RF, Buede DM. Theoretical framework for the continuous early validation (CEaVa) method. *Syst. Eng.* **2002**, *5*, 223–241. doi:10.1002/sys.10022.
27. Jabareen Y. Building a conceptual framework: Philosophy, definitions, and procedure. *Int. J. Qual. Methods* **2009**, *8*, 49–62. doi:10.1177/160940690900800406.

28. Glaser B, Strauss A. *The Discovery of Grounded Theory: Strategies for Qualitative Research*; Routledge: New York, NY, USA, 2017; p. 282, doi:10.4324/9780203793206.
29. Wooldridge M. *An Introduction to Multiagent Systems*, 2nd ed.; John Wiley & Sons: Chichester, UK, 2009; p. 484.
30. Giret A, Trentesaux D, Salido MA, Garcia E, Adam E. A holonic multi-agent methodology to design sustainable intelligent manufacturing control systems. *J. Clean. Prod.* **2017**, *167*, 1370–1386. doi:10.1016/j.jclepro.2017.03.079.
31. Palau AS, Dhada MH, Parlikad AK. Multi-agent system architectures for collaborative prognostics. *J. Intell. Manuf.* **2019**, *30*, 2999–3013. doi:10.1007/s10845-019-01478-9.
32. Wellsandt S, Klein K, Hribernik K, Lewandowski M, Bousdekis A, Mentzas G, et al. Hybrid-augmented intelligence in predictive maintenance with digital intelligent assistants. *Annu. Rev. Control* **2022**, *53*, 382–390. doi:10.1016/j.arcontrol.2022.04.001.
33. Shen W, Hao Q, Yoon HJ, Norrie DH. Applications of agent-based systems in intelligent manufacturing: An updated review. *Adv. Eng. Inform.* **2006**, *20*, 415–431. doi:10.1016/j.aei.2006.05.004.
34. Leitão P, Karnouskos S. *Industrial Agents: Emerging Applications of Software Agents in Industry*; Elsevier: Amsterdam, Netherlands, 2015; p. 455. doi:10.1016/C2013-0-15269-5.
35. Soon GK, On CK, Anthony P, Hamdan AR. A review on agent communication language. *Lect. Notes Electr. Eng.* **2019**, *481*, 481–491. doi:10.1007/978-981-13-2622-6_47.
36. Harris S, Seaborne A. SPARQL 1.1 Query Language. World Wide Web Consortium (W3C). 2013. Available online: https://www.w3.org/TR/sparql11-query/ (accessed on 12 November 2024).
37. Stokel-Walker C, Van Noorden R. What ChatGPT and generative AI mean for science. *Nature* **2023**, *614*, 214–216. doi:10.1038/d41586-023-00340-6.
38. Brooks R. A robust layered control system for a mobile robot. *IEEE J. Robot. Autom.* **1986**, *2*, 14–23. doi:10.1109/JRA.1986.1087032.
39. Bettosini I, Clavelli A, Barnech GT, Visca J, Benavides F. Torocó: A subsumption architecture implementation. In *Proceedings of the 8th International Conference on Automation, Robotics and Applications, Prague, Czech Republic, 18–20 February 2022*; IEEE: New York, NY, USA, 2022; pp. 27–32. doi:10.1109/ICARA55094.2022.9738521.
40. Rao AS, Georgeff MP. BDI agents: From theory to practice. In Proceedings of the First International Conference on Multiagent Systems, San Francisco, CA, USA, 12–14 June 1995; Volume 95, pp. 312–319.
41. Jang M, Yoon Y, Choi J, Ong H, Kim J. A structured prompting based on belief-desire-intention model for proactive and explainable task planning. In Proceedings of the 11th International Conference on Human-Agent Interaction, New York, NY, USA, 4 December 2023; pp. 375–377. doi:10.1145/3623809.3623930.
42. Ferguson IA. TouringMachines: An architecture for dynamic, rational, mobile agents. In *Technical Report No. UCAM-CL-TR-273*; University of Cambridge, Computer Laboratory: Cambridge, UK, 1992. doi:10.48456/tr-273.
43. Xu J, Wang L, Kou Q, Fang T, Dan Y, Zhou L, et al. Real-time behaviour decision of mobile robot based on the deliberate/reactive architecture. *Int. J. Innov. Comp. Inform. Control* **2022**, *18*, 1163–1180. doi:10.24507/ijicic.18.04.1163.
44. Shen ZQ, Gay R, Tao X. Goal-based intelligent agents. *Int. J. Inform. Tech.* **2003**, *9*, 19–30.
45. Dulek B, Efendi E, Varshney PK. Behavioral utility-based distributed detection with conditionally independent observations. *IEEE Trans. Sign. Proc.* **2024**, *72*, 3717–3730. doi:10.1109/TSP.2024.3439732.
46. Paccagnan D, Chandan R, Marden JR. Utility and mechanism design in multi-agent systems: An overview. *Ann. Rev. Control* **2022**, *53*, 315–328. doi:10.1016/j.arcontrol.2022.02.002.
47. Zhang J, Bedi AS, Wang M, Koppel A. Multi-agent reinforcement learning with general utilities via decentralized shadow reward actor-critic. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 27 February 2022; Volume 36, pp. 9031–9039. doi:10.1609/aaai.v36i8.20887.
48. Russell S. Learning agents for uncertain environments. In Proceedings of the 11th Conference on Computational Learning Theory, Madison, WI, USA, 24–26 July 1998; pp. 101–103. doi:10.1145/279943.279964.
49. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing Atari with deep reinforcement learning. *arXiv* **2013**. doi:10.48550/arXiv.1312.5602.
50. Li C, Zheng P, Yin Y, Wang B, Wang L. Deep reinforcement learning in smart manufacturing: A review and prospects. *CIRP J. Manuf. Sci. Tech.* **2023**, *40*, 75–101. doi:10.1016/j.cirpj.2022.11.003.
51. Newell A. The knowledge level. *Art. Int.* **1982**, *18*, 87–127. doi:10.1016/0004-3702(82)90012-1.
52. Jackson P. *Introduction to Expert Systems*; Addison-Wesley: Boston, MA, USA, 1990; p. 526.
53. Russell SJ, Norvig P. *Artificial Intelligence: A Modern Approach*, 3rd ed.; Pearson Education Limited: Harlow, UK, 2016; p. 1132.
54. Wrona Z, Buchwald W, Ganzha M, Paprzycki M, Leon F, Noor N, et al. Overview of software agent platforms available in 2023. *Information* **2023**, *14*, 348. doi:10.3390/info14060348.
55. Dong D. Enabling an autonomous agent sharing its minds, describing its conscious contents. *Cogn. Syst. Res.* **2023**, *80*, 103–109. doi:10.1016/j.cogsys.2023.03.001.