**Reflection and E-Portfolio Report for Intelligent Agents Module**

**Student ID:** 32483

**Module:** Intelligent Agents

**GitHub Repository URL:** <https://rmalnaqbi.github.io/Rayyan-eportfolio/Intelligent%20Agents.html>

**Date:** 20 October 2025

**Reflective Of Intelligent Agents Module**

**Student ID:** 32483

**Module:** Intelligent Agents

**GitHub Repository URL:** <https://rmalnaqbi.github.io/Rayyan-eportfolio/Intelligent%20Agents.html>

**Date:** 20 October 2025

**1. Introduction**

My learning experience during the Intelligent Agents module is reviewed in the reflection. It talks about how the course altered my understanding of software development, Python programming, and agent-based systems. GitHub also helped me become a better team player, and the module improved my understanding of the ethical and professional factors in the design of autonomous systems.

The way the reflection is structured follows the Rolfe et al. (2001) model. What? What the heck? Now What? - to evaluate what I have learned, why it matters, and how I will use these lessons in the upcoming projects.

**2. Description of Experience (What?)**

I had little knowledge of agent-based systems and a very broad understanding of artificial intelligence before taking this module (Russell and Norvig, 2021). I learned in the first unit that intelligent agents are self-governing software applications that have the ability to respond, reason, and interact in dynamic settings. The ideas that formed the foundation of my knowledge were autonomy, initiative, and social interaction.

I studied reactive, deliberative, and hybrid agent architectures during the first few units. I learned how each architecture helps agents make decisions and solve problems. I was able to connect theoretical ideas with practical development through the use of small-scale Python examples. I also realized that in order to enable an agent to act autonomously and react to environmental changes, programming logic is crucial and needs to be well-structured.

The group project was one of the module's most crucial components. Working in a team to develop an agent-based system gave me real-world experience with version control and teamwork. The code, roles, and progress were all managed via GitHub. Maintaining the repository, checking commits, and ensuring code consistency were my duties (GitHub, Inc., 2025). I learned to appreciate the importance of discipline and teamwork in software projects as a result of this.

My understanding of how agents interact and how activities are coordinated using set protocols has expanded as a result of subsequent courses. I also found it intriguing how I could use Python to translate theoretical models of cooperation into real simulations.

**3. Introspection and Emotional Response (So What?)**

At first, I had issues with the technical content. These terms were difficult to understand because they were agent communication languages, hybrid models, and logical reasoning. Syntax and logical errors made the early project programming experiences discouraging. However, these setbacks taught me perseverance and problem-solving techniques, which are critical abilities for any developer.

As the weeks passed, my confidence grew. Working with other team members on GitHub was more efficient, and I started to enjoy debugging and improving our code. Seeing our system gradually take shape was satisfying. It was very similar to the real world of software development, where collaboration, communication, and consistency are the only ways to advance development.

I was able to move from uncertainty to confidence thanks to the emotional experience. I came to the realization that reflecting and feeling frustrated are both learning processes based on Gibbs' Reflective Cycle. Every challenge, whether cooperative or technical, aided in my development as a reflective practitioner.

The most significant factor was an ethical consideration in the design of autonomous systems. Additionally, I learned that developers are accountable for the behavior of intelligent agents and that they must be transparent (Wooldridge, 2020). I was able to comprehend that technology design is a technical and ethical endeavor thanks to the argument regarding accountability and fairness (Johnson, 2020). This information changed the way I thought about programming: it's a system that can affect people and society, not just a set of codes.

My online professionalism was also improved by using GitHub during the module. I learned how to maintain repositories, write my work in an understandable style, and use version control responsibly (Van Rossum and Drake, 2009). The platform made me feel like I owned my contributions and encouraged teamwork. Possessing a well-structured repository demonstrated industry professional development activities.

**4. Lesson Learned and Future Application (Now What?)**

My technical and professional personality has been shaped by the module. It made it possible for me to progress from a theory-focused student to a responsible developer capable of creating and managing intelligent systems.

**Technical Skills:**

I gained a solid understanding of both the Python programming language and the idea of creating reusable and modular code. I gained knowledge on how to logically integrate model interactions with agency behavior. To create more sophisticated agent-based models in the future, I'll take into account models like Mesa.

**Software Development**: This phase entails developing efficient software to guarantee a business runs smoothly. Software Development: This stage focuses on creating effective software that lets a business operate at its own speed.

Through GitHub, I have learned the importance of documentation, version control, and teamwork. I can now effectively manage the repositories, review code, and resolve conflicts. These abilities will be useful in the classroom and in situations involving the workplace where accountability and teamwork are essential.

**Professional Awareness**: I am developing ethical awareness and professional awareness as well.

**Ethical and Professional Awareness**: The module improved my knowledge of professional responsibility. I also learned that smart systems may have an influence on the outcome of real-world relationships, and developers should pay attention to the ethical, social, and legal consequences of the design process. I will keep using the ethical reasoning to all the projects I will be working on.

**Future Goals:**

To keep going forward, I need to accomplish a number of objectives:

1. Continue using Python by contributing to the open source community.

2. Apply agile collaborative coding techniques.

3. Increase knowledge of agent-based simulation tools.

4. Regularly reflect using the Rolfe model to evaluate learning and development.

Taking such actions will help me continue to be technically and morally proficient.

**5. Conclusion**

Looking back at this module, I can see quite clearly the way my technical competence, teamwork, and professional mind have been shaped. I started with less knowledge of intelligent systems and now I have learned how to design code and evaluate intelligent systems using Python.

I have learned through project management on GitHub that organization and teamwork are essential for software development success. At the same time, the module's ethical component taught me that creating intelligent systems is a duty that needs to be open and honest.

**References:**

* Johnson, D.G. (2020) *Computer Ethics*. 5th ed. Pearson Education. Available at: [https://mrce.in/ebooks/Computer%20Ethics%204th%20Ed.pdf](https://mrce.in/ebooks/Computer%20Ethics%204th%20Ed.pdf?utm_source=chatgpt.com) (Accessed: 18 October 2025).
* GitHub, Inc. (2025) *Collaborating with Issues and Pull Requests*. Available at:<https://docs.github.com/en/issues> (Accessed: 18 October 2025).
* Russell, S. and Norvig, P. (2021) *Artificial Intelligence: A Modern Approach*. 4th ed. Harlow: Pearson Education. Available at: [https://aima.cs.berkeley.edu/](https://aima.cs.berkeley.edu/?utm_source=chatgpt.com) (Accessed: 18 October 2025).
* Wooldridge, M. (2020) *An Introduction to MultiAgent Systems*. 3rd ed. Hoboken, NJ: John Wiley&Sons. Available at:https://uranos.ch/research/references/Wooldridge\_2001/TLTK.pdf (Accessed: 18 October 2025).
* Van Rossum, G. and Drake, F.L. (2009) *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. Available at: [https://www.ecampus.com/python-3-reference-manual-van-rossum/bk/9781441412690](https://www.ecampus.com/python-3-reference-manual-van-rossum/bk/9781441412690?utm_source=chatgpt.com) (Accessed: 18 October 2025).
* University of Edinburgh (no date) Reflection Toolkit. The University of Edinburgh. Available at: <https://www.ed.ac.uk/reflection> (Accessed: 18 October 2025).

**E-Portfolio Report**

**Student ID:** 32483

**Module:** Intelligent Agents

**GitHub Repository URL:** <https://rmalnaqbi.github.io/Rayyan-eportfolio/Intelligent%20Agents.html>

**Date:** 20 October 2025

**1. Introduction**

This report details the creation and submission of my e-Portfolio for the University of Essex's Intelligent Agents module. My knowledge of mechanics, how to approach real work in Python, and my experience working in a virtual collaborative team are all evidenced in my GitHub-hosted e-portfolio, which documents my knowledge development over the course of the twelve courses.

With an emphasis on architectures, communication protocols, hybrid approaches, and ethical considerations, the module explored the design, implementation, and assessment of agent-based systems. Using the e-Portfolio, I have subjectively assessed my development of critical skills, such as problem-solving, system design, coding proficiency, and responsible AI practice.

I organized my development during the course period using the Kerb sphere of experiential learning (Kolb, 1984) and a reflective team of Rolfe et al. (2001). While technical achievements make up the majority of the e-Portfolio, it is also possible to see the integration of professional values, teamwork, and ethics in the digital sphere.

**2. E-Portfolio Work description**

This is my e-Portfolio, which is located in the Intelligent-Agents repository on GitHub. The weekly tasks, group discussions and work, and project work are all neatly organized in this log. Each folder reflects a module's units and contains markdown (.md) files, weekly learning results summaries, code snippets, screen captures, and evaluation notes.

Each unit had its own link, and the module's general overview was provided in the README.md file. This made navigating around it simple and straightforward. They were separated into sections based on themes:

**Unit 1–4:** Characteristics of Agents and the Foundations of Their Architectures and Logics.

**Units 5-8:** Natural language processing, ontologies, and agent communication.

**Units 9–12:** Combining intelligent agents and adaptive algorithms with deep learning.

includes theoretical synopses and real-world Python applications for each section. I worked on some simulation exercises in Jupiter Notebooks, such as experiments with ontology-based decision systems, agent prototypes with reactive and deliberative agents, and agent communication and message-passing protocols.

There were screenshots, annotated code snippets and comments by peers, which were also used to show teamwork and contribution within group work. The e-Portfolio is therefore not just a document of learning but it is also a professional demonstration of cooperative software development.

**3. Key Learning Outcomes**

**3.1 Python and Comparable Development**

One of the most significant results of this module was the development of my technical capability. I was able to distinguish between intelligent agents and conventional programs by using the introductory lessons. Intelligent agents possess autonomy, reactivity, proactivity, and socialization skills (Wooldridge, 2021). My understanding was improved by converting these theoretical ideas into working code.

In Python, I wrote simple reflex agents, model-based agents followed by goal-based designs on the models. Unit 3 involved me learning about Belief-Desire-Intention (BDI) architecture that enabled agents to strike a balance between dynamic decision-making and long-term goals. Use of Python libraries e.g. spade and Mesa helped me in simulating multi-agent environment, implement communication protocols, and visualization of agent interactions.

My mental software engineering mindset was facilitated by this application practice, particularly in the areas of debugging, modularity, and algorithm reasoning. Intelligent agents must integrate perception, reasoning, and action, claim Russell and Norvig (2022). Through trial and error, I developed agents that could adjust to changing environments, which emphasizes the module's significance in relation to risk and uncertainty management.

I became more acquainted with the actual working procedures by using Github teamwork. Commits, pull requests, and issue tracking helped me create the agile development cycle. In order to ensure that my coding was transparent and reproducible, Pressman and Maxim (2020) advised professional engineers to follow this practice.

**3.2 Virtual Collaboration and Teamwork**

Establishing communication and coordination skills was greatly aided by the teamwork task element. We worked on an agent-based resource allocation system as part of a distributed virtual team. Each of the project members was given a specific role, such as documenter, ontology designer, coordinator, or developer. At GitHub, my most frequent contributions were in Python and version control.

Our working tools also included GitHub, Microsoft Teams, and shared Google Docs, where we proposed the design and kept each other informed of its development. In line with Belbin's (2010) theory on team roles, which emphasizes skill balance in productive teams, this is how technical and interpersonal skills were developed.

Version blending and role duplication occasionally led to conflict, but we were able to resolve these issues through constructive communication by using Gibbs' (1988) reflective framework. To facilitate productive collaboration, we have been trained to use pull requests, assign tasks transparently, and properly document changes made. My understanding of professional software development teamwork and virtual project management has improved as a result of this experience.

The potential final product, a hybrid agent system prototype, demonstrated not only technical integrity but also non-technical abilities like flexibility, leadership, and negotiation. The combination of reactive and deliberative layers in our design shows that high-tech architecture can offer a compromise between flexibility and performance (Ferber, 1999).

Every individual has different values to uphold and standards to take into account.

**3.3 Professional and Ethical Awareness**

Each person has their own standards to adhere to and norms to take into account.

Social, legal, and ethical considerations were integrated throughout the entire module. We looked at Unit 10 where the author talked about how using AI systems like agents without taking responsibility for their creation can lead to bias or privacy issues. When we were working on group projects, this was more relevant when it came to simulated decision data.

All of the coding and documentation was done ethically with respect to authorship and data transparency, in compliance with the University of Essex Academic Integrity Policy (2023). To make sure that our system design, which was based on an agent, was also just and responsible, we also took into account the four AI ethics principles offered by Floridi and Cowls (2019): beneficence, non-malfeasance, justice, and explicability.

I learned that the development of intelligent agents is more than just a technical undertaking with the aid of such a moral lens. It addresses moral reasoning and taking social influence into account. These perspectives helped me comprehend the professional practice in AI and software development, where it is required to adhere to IEEE standards like Ethically Aligned Design (2021).

**4. Reflection and critical Thinking**

I will use the example of developing inter-agent communication with ontologies, where I saw how human beings reflect miscommunication. This improved my perception of the design clarity and semantic accuracy even more. Similar to this, additional classes that focused on adaptive-based algorithms helped me understand the relationship between agents and machine learning, demonstrating how learning systems and feedback can be used to generate intelligence.

The experience also had an impact on my professional identity. I now see a software development process as a morally and team-building exercise rather than as a distinct technical one. Kolb (1984) asserts that students generate new knowledge when they are able to modify their experiences through reflection during the learning process, which is made possible in this module by team-level documentation and evaluation in an e-portfolio.

**5. Future Uses and Future Development**

software development and AI in the future. I plan to further my education and advance my career in order to develop them.

My second goal is to simulate an agent-based energy manager simulation using reinforcement learning and the Python Mesa library to carry out adaptive control. In order to solve the sustainability problem, this will incorporate the concepts from Units 9–11 on communication, deep learning, and adaptive algorithms.

In order to enhance my employability even further, I have established the following SMART objectives:

* **Specific:** Create and showcase a GitHub project that exemplifies a hybrid agent system that takes ethical considerations into consideration.
* **Measurable:** By March 2026, modify and test decision-making units and communications specifications.
* **Achievable**: Spend 4 hours per week on Python coding as an agent, and documenting the project.
* **Relevant**: Make the project match the Industry 4.0 applications, specifically, in the field of IoT and automation.
* **Time limit**: hand over at the expiry of the following academic term.

Besides this, I intend to donate to open-source communities that are researching on intelligent agents. Engaging in GitHub discussions will enable me to work and cooperate with people around the world and put the lessons into practice.

In the long term, I want to apply to the AI development or automation design that will be aimed at human collaboration, but not the elimination. Because, as Russell (2019) puts it, the implementation of AI in accordance with the human values is the key to the sustainable innovation, I have implemented this core concept into my professional vision.

**6. Conclusion**

An academic and professional accomplishment was the development of the Intelligent Agents module e-Portfolio. It enhanced my understanding of Python, solidified my understanding of agent-based systems, and bolstered my value of ethics, collaboration, and lifelong learning.

I became familiar with the concept of how intelligent agents might be created to behave responsibly in an uncertain environment by utilizing the partnership with GitHub and critical thinking regarding theoretical frameworks. The process combined this idea with the idea that effective AI development necessitates not only coding expertise but, more crucially, ethical accountability, communication, and introspection. My experience has prepared me to apply these competencies in real-world scenarios where fair, transparent, and human-centered intelligent systems are required.

**References:**

* Finin, T., Labrou, Y. and Mayfield, J. (1994) KQML as an agent communication language. University of Maryland. Available at: <https://www.cs.umbc.edu/kqml/> (Accessed: 18 October 2025).
* Belbin, R. M. (2010). Team Roles at Work (2nd ed.). Oxford: Butterworth-Heinemann. Available at:[https://www.taylorfrancis.com/books/mono/10.4324/9780080963242/team-roles-work-meredith-belbin Taylor & Francis+2PagePlace+2](https://www.taylorfrancis.com/books/mono/10.4324/9780080963242/team-roles-work-meredith-belbin?utm_source=chatgpt.com) (Accessed: 18 October 2025).
* Ferber, J. (1999). Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Harlow: Addison-Wesley. Available at: [https://archive.org/details/multiagentsystem0000ferb Internet Archive+2BibSonomy+2](https://archive.org/details/multiagentsystem0000ferb?utm_source=chatgpt.com) (Accessed: 18 October 2025).
* Floridi, L. & Cowls, J. (2019). “A Unified Framework of Five Principles for AI in Society.” Harvard Data Science Review, 1(1). Available at: Available at:<https://doi.org/10.1162/99608f92.8cd550d1> [Harvard Data Science Review](https://hdsr.mitpress.mit.edu/pub/l0jsh9d1?utm_source=chatgpt.com) (Accessed: 18 October 2025).
* Guadamuz, A. (2023) Generative AI and copyright: The challenges of ownership and originality. Journal of Intellectual Property Law C Practice. Available at: <https://academic.oup.com/jiplp/article/18/5/393/7145741> (Accessed: 18 October 2025).
* IEEE Standards Association. (2021). Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems. Available at: [https://standards.ieee.org/wp-content/uploads/import/documents/other/ead\_v2.pdf IEEE Standards Association](https://standards.ieee.org/wp-content/uploads/import/documents/other/ead_v2.pdf?utm_source=chatgpt.com) (Accessed: 18 October 2025).
* Kolb, D. A. (1984). Experiential Learning: Experience as the Source of Learning and Development. Englewood Cliffs, NJ: Prentice Hall. Available from: https://www.researchgate.net/publication/235701029\_Experiential\_Learning\_Experience\_As\_The\_Source\_Of\_Learning\_And\_Development (Accessed: 18 October 2025).