E-PORTFOLIO SUBMISSION

Module 5: Intelligent Agents

MSc Artificial Intelligence

University of Essex Online

Pavlos Papachristos

Student ID: 24589

ePortolio link: https://spike2025-art.github.io/eportfolio-uoe/

Table of Contents

PART 1: E-PORTFOLIO DESCRIPTION

1.1 Introduction and Purpose

This e-portfolio documents my learning journey through Module 5: Intelligent Agents, presenting evidence of theoretical understanding, practical implementations, and collaborative project work. The portfolio demonstrates achievement of module learning outcomes through concrete artifacts including agent implementations, communication protocols, natural language processing tools, and a multi-agent cybersecurity platform.

The submission serves three purposes: a) demonstrating learning outcome achievement through tangible evidence, b) providing critical reflection on the learning process and insights gained and c) documenting individual contributions to team projects. The portfolio encompasses both technical artifacts and metacognitive reflection on the learning experience.

ePortfolio link

The ePortolio link: https://spike2025-art.github.io/eportfolio-uoe/

1.2 Portfolio Structure and Key Work

The e-portfolio is organized on my GitHub Pages website (https://spike2025-art.github.io/eportfolio-uoe/) under Module 5, presenting work chronologically across units (see figure 1).

Unit 1: Collaborative Discussion - The Rise of Agent-Based Systems

This unit explores the increasing adoption of agent-based systems in modern organizations through a collaborative forum discussion. The initial post examines why traditional centralized systems are being replaced by distributed, intelligent agent architectures that offer greater flexibility and resilience in complex, fastchanging environments. The discussion includes responses to the forum that extend the conversation into the role of reinforcement learning in multi-agent coordination.

The key topics covered are the decentralized intelligence distribution across autonomous agents, the applications in supply chain management and manufacturing and the integration with AI advances including machine learning and large language models (LLMs).

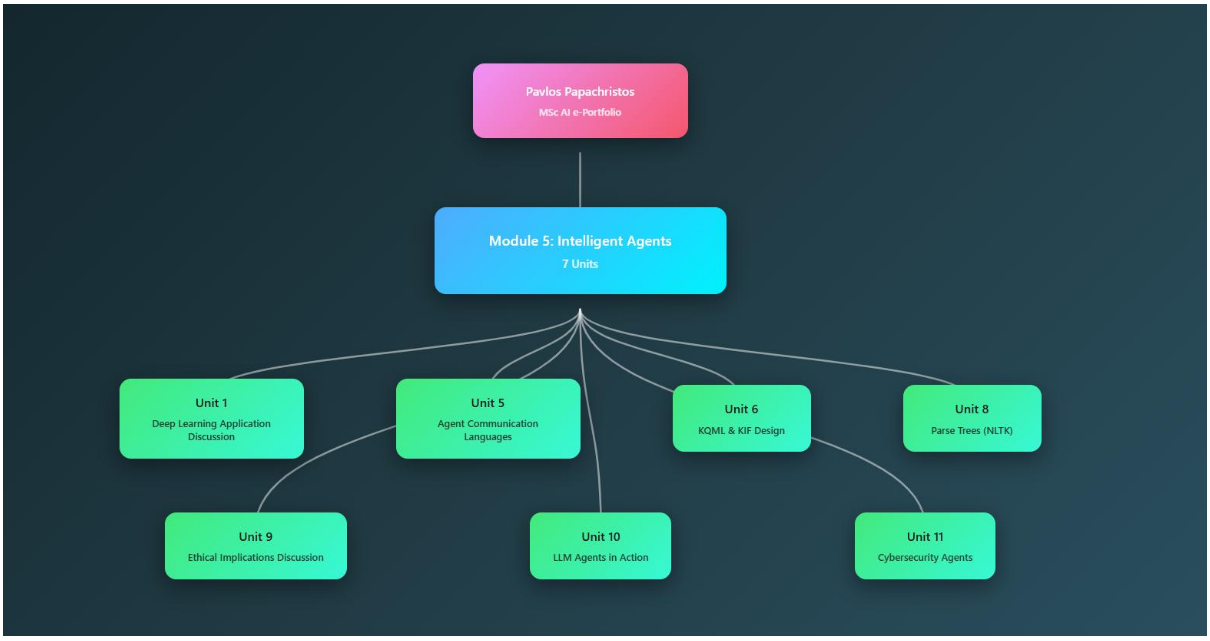

Figure 1: Module 5 Interactive Structure (click CTRL+ Image to explore online)

Unit 5: Collaborative Discussion - Agent Communication Languages

A critical exchange about how agents communicate. I compared Agent Communication Languages (ACLs) such as KQML and FIPA ACL with RESTful APIs and emerging large language model interfaces. I argued that communication should be context-driven: formal semantics for accountability and auditability, and flexible language when experimentation or scalability is the main goal.

Unit 6: Constituency-Based Parse Tree Analysis

This unit provides comprehensive coverage of syntactic parsing using constituency-based parse trees, a fundamental technique in Natural Language Processing (NLP). The material progresses from basic tree structures to complex ambiguity resolution.

The document progresses from simple subject-verb-object constructions to ditransitive verbs and ultimately to structures requiring contextual interpretation and demonstrates why NLP remains challenging despite advances in AI.

Figure 2: Module 5 – KQML_KIF – Message Flow Structure

Unit 6 includes also the development team project report that documents the design and implementation of a Multi-Agent Email Forensics System using Python. The system demonstrates practical application of agent-based architectures to cybersecurity.

Unit 8: Creating Parse Trees (NLTK)

This exercise used NLTK to generate parse trees illustrating syntactic structure and ambiguity. By exploring attachment ambiguity, it became clear how even simple sentences can produce multiple valid interpretations. The task reinforced the difficulty of handling natural language in computational reasoning systems.

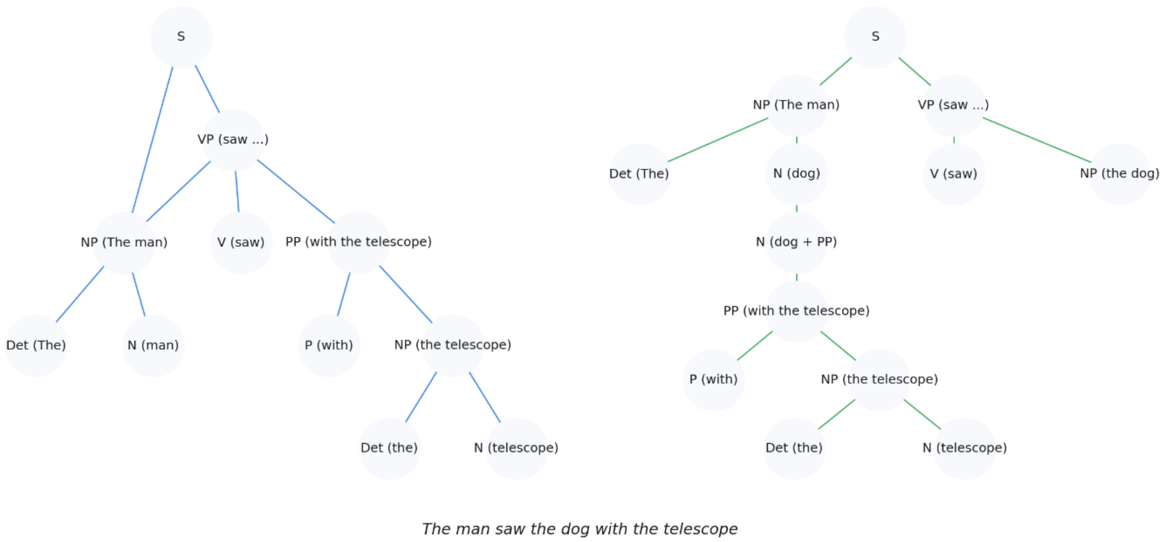

Figure 3: Module 5 – Parse Tree Ambiquity

Unit 9 — Collaborative Discussion: Ethical Implications of Deep Learning

This collaborative discussion explored the ethical tensions raised by generative AI: authenticity, bias, ownership, and accountability. I argued for proactive governance, emphasising transparency and stakeholder involvement as key to aligning AI with public good rather than purely commercial interests.

Unit 10 — Deep Learning in Action: LLM Agents

Analysed the role of large language models in agent design. I reflected on their advantages—rich linguistic reasoning and contextual flexibility—alongside risks like hallucination and prompt injection. I proposed combining structured communication for safety-critical layers with open-ended natural language for adaptive reasoning.

Unit 11 — Project Presentation - Intelligent Agents in Action (Cybersecurity)

The presentation synthesizes the practical implementation from Unit 8, demonstrating the complete email forensics system to stakeholders. The system used multiple cooperating agents for malware detection, evidence collection, and event coordination.

The key characteristics of the system include:

- System architecture — designed a modular, event-driven structure with SQLite for lightweight evidence storage and shared threat intelligence.
- Malware detection agent — implemented recursive file analysis, hashing, heuristic scoring, and automatic quarantine. Optimised performance from 200 to over 1,200 files per second.
- Testing and integration — developed unit and integration tests, validated inter-agent communication, and maintained documentation.

Presentation_pp
24589_fin.pptx

PART 2: CRITICAL REFLECTION

2.1 Learning Journey and Evolution

When I have started studying for Module 5, I thought I had figured out what the AI agent are. I had done Python programming in earlier modules and picked up general AI concepts, so agents seemed like just "programs that run on their own." Looking back now, that is almost embarrassingly simple. I had no real sense of the theoretical depth behind agent architectures, and communication protocols were basically foreign territory to me.

What has surprised me most is how this module changed not just what I know, but how I think. I used to see agents as standalone software, but now I understand they're part of intricate coordinated systems. The communication between agents is nowhere near as straightforward as I assumed. I've moved from obsessing over individual agent intelligence to realizing the key often is at the system level, where behaviours emerge from interactions. The ethical side became much more real after working on the cybersecurity project.

I expected coding challenges, and sure, there was plenty of that. But the module threw me off balance. I found myself wrestling with philosophical questions about what agency means—speech act theory material I never expected reading in a tech module. The formal methods overwhelmed me initially. The deployment side—putting these systems into the real world it is very different from just getting code to work.

The biggest revelation to me is the enormity of the field. What looked like a contained technical area turned out to be this sprawling discipline with so many architectures, communication theories, coordination approaches, planning methods.

2.2 Understanding AI Agents

One thing that is crystallized for me is understanding what agents do and why they matter. They bring modularity—each agent handles its own responsibilities. What is neat is when you need to change something, you're not ripping through an entire codebase. Changes stay contained, making maintenance easier.

Scalability is another win. Instead of one massive system doing everything, multiple agents distribute the workload. If one agent crashes? The system does not necessarily go down with it. There is the element of degradation that is elegant when is working.

Rather than forcing processes into artificial technical frameworks, you can have agents reflecting real organizational roles. A pricing agent sets prices, a risk agent evaluates transactions, a compliance agent ensures regulatory adherence. They negotiate and coordinate almost like people in an organization would.

The coordination mechanisms took longer to understand. The KQML/KIF implementation showed me formal communication—standardized message types with shared knowledge representation. With the cybersecurity platform, we used event-driven coordination. Agents published events, the platform handled routing. Much looser coupling, more flexible. Both approaches work, just for different situations.

2.3 Challenges Encountered

The KQML/KIF implementation exposed serious gaps in my understanding. When started getting message formatting right, tracking conversation state, parsing those KIF expressions with proper first-order logic syntax, handling errors when things inevitably broke—it become apparent that this was way more involved than expected.

Formal approaches give you semantic clarity and let you reason formally. But they come with huge development overhead that simpler alternatives do not have. I kept waiting for a clear answer about when to use which approach, but it never came. Turns out it depends on context. Need formal verification? Use the heavyweight stuff. Need to ship quickly? Go simpler. There's no universal "best" answer, which was frustrating but probably more realistic.

The NLP work brought another challenge—ambiguity is everywhere. "The man saw the dog with the telescope"? Is the man using the telescope, or does the dog have it? We humans figured it out instantly, but the computational system just couldn't. That makes me appreciate how much of intelligence involves dealing with uncertainty and ambiguity, not just executing algorithms. Real problems often do not have single correct answers—they're fundamentally ill-defined.

The team project brought yet another set of challenges. Building individual components was challenging, but integration was where things got more demanding. We'd designed careful interfaces, documented everything, thought we had it figured out. Then we started connecting agents and issues popped up in several areas. The time spend debugging integration issues almost equals the writing original code.

That has highlighted the importance of systems thinking. It is not enough to make sure each component works alone. You must think about how they interact, how errors propagate, how timing affects things. The gap between "individual parts work" and "system works as a whole" can be surprisingly wide.

2.4 The Expanding Landscape

One thing that has been both exciting and daunting, it is realizing just how much is out there. For almos any challenge—communication, coordination, knowledge representation, planning, learning—multiple established approaches exist with extensive research. KQML versus FIPA ACL versus REST APIs. Rule-based versus learning-based reasoning. The options just keep going.

On one hand, that is great. Whatever problem you're facing, there's likely an established solution. On the other hand, as a learner building coherent understanding it is a lot. How do you decide what to study deeply versus understand superficially and how to evaluate competing approaches.

At the moment there is a cosmogonic explosion of LLM-based agent. Systems like AutoGPT demonstrate capabilities—understanding natural language, using tools, planning dynamically—those previous systems only achieved through massive hand-crafted knowledge bases. These developments happened in real-time during this module. Papers on ReAct and Tree of Thoughts are already essential reading (Yao et al., 2023). It is remarkable how fast things move.

This rapid pace reinforces something: AI development requires continuous learning just to stay current. The classical concepts we studied provide crucial foundations. But I can't stop there. I also need to track what is happening with LLM-based agents because that is clearly where the field is heading.

2.5 Team Collaboration Insights

Working with a distributed team across three time zones taught me things solo work would not. We held regular updates on what we were doing, what blocked us, what we had tackle next. But we also had weekly video calls for discussions and architectural decisions.

Code reviews ended up more valuable than expected. And they have become the main knowledge transfer mechanism. When I reviewed someone's code and suggested improvements, I wasn't just helping them—I was solidifying my own understanding through clear explanations.

What struck me was the team synergy. Expertise brough together in various fields such as software architecture, visualization, database design, documentation. That diversity meant our collective output genuinely exceeded what any of us could produce individually. When teams work well, that synergy is kind of remarkable.

2.6 Conclusions and Forward Path

Looking back at where I started versus where I am now—it is surprising how much has changed. I have now a completely different understanding of architectural patterns, communication mechanisms, coordination approaches. It is not that I know everything now—far from it. But how I think about these systems has fundamentally shifted.

The progression from theory to hands-on implementation was key. You can spend hours reading about architectures but building them exposes all these little details and weird edge cases that textbooks skip or make sound trivial.

The cybersecurity platform showed clearly that coordinated multi-agent systems can achieve things no single agent—no matter how sophisticated—could pull off alone. That emergent capability from coordination is something else.

I am now better at evaluating different technical approaches now—not just picking what sounds good but thinking through trade-offs systematically. I can look at multiple sources that contradict each other and figure out what is going on. I am comfortable with the fact that I' am never going to "finish" learning this stuff. Mastery takes years.

These meta-learning skills matter more than knowing specifics of any framework, especially in a field moving as fast as AI. Today's hot framework might be obsolete in two years. But knowing how to learn continuously, evaluate new developments critically, integrate innovations without chasing hype—those capabilities have staying power.

The timing of this module has been interesting. We're at this inflection point where LLM-based agents demonstrate capabilities that used to require massive custom engineering. It is exciting, but also a bit humbling—the landscape keeps shifting faster than I can keep up with sometimes. At least the foundations I've built give me context for engaging with new developments critically instead of just getting swept along.

As for what is next, multi-agent reinforcement learning attract me most. I am fascinated by how agents learn optimal strategies when interacting with other learning agents, how everyone's strategy is evolving simultaneously and how does converge.

Formal verification methods are another area I want to expand. As these systems get deployed in critical contexts being able to prove they are safe becomes crucial. I want to build more sophisticated applications too focusing on their application in the financial sector.

Staying current with LLM-based agent research feels important, although separating genuine breakthroughs from hype is challenging. I would like to contribute to open-source agent frameworks if I can find the right projects. Partly for practical experience and to give back to the community. I've learned so much from open-source projects others have built.

This module opened a door I didn't even know was there. What I found was this vast, fascinating field I've only just started exploring. The journey's only beginning. There's so much more to cover, so many directions I could go and I am looking forward to finding it out what comes next.

REFERENCES

Finin, T., Labrou, Y. and Mayfield, J. (1997) 'KQML as an Agent Communication Language', in Bradshaw, J.M. (ed.) Software Agents. Cambridge, MA: MIT Press, pp. 291-316.

Jennings, N.R., Sycara, K. and Wooldridge, M. (1998) 'A Roadmap of Agent Research and Development', Autonomous Agents and Multi-Agent Systems, 1(1), pp. 7-38.

Jurafsky, D. and Martin, J.H. (2023) Speech and Language Processing. 3rd edn. Stanford: Stanford University Press.

Rao, A.S. and Georgeff, M.P. (1995) 'BDI Agents: From Theory to Practice', in Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95). San Francisco, CA: AAAI Press, pp. 312-319.

Russell, S.J. and Norvig, P. (2020) Artificial Intelligence: A Modern Approach. 4th edn. Hoboken, NJ: Pearson.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. and Cao, Y. (2023) 'ReAct: Synergizing Reasoning and Acting in Language Models', in Proceedings of the International Conference on Learning Representations (ICLR 2023).

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. and Cao, Y. (2022) 'ReAct: Synergizing Reasoning and Acting in Language Models', arXiv preprint arXiv:2210.03629. Available at: https://arxiv.org/abs/2210.03629 (Accessed: 18 October 2025).