AI in Society and Public Services

Session 6: From Chatbots to Autonomous Agents

Mário Antunes

February 04, 2026

Universidade de Aveiro

Table of Contents i

AI in Society and Public Services

Session details i

Session 6: From Chatbots to Autonomous Agents

Duration: 3 Hours

Instructor: Mário Antunes

Session details ii

Scan the QR code below to access all slides, code examples, and resources for this workshop.

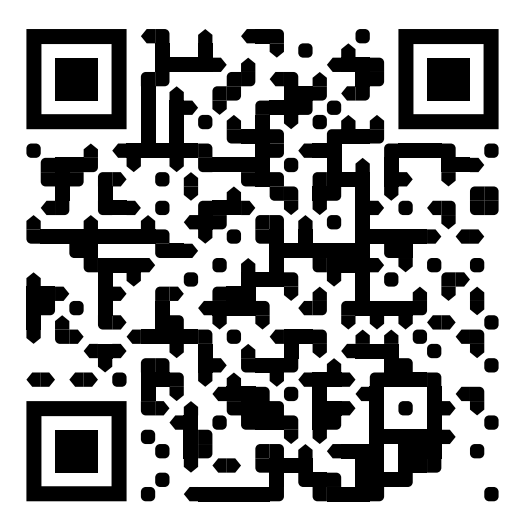

Figure 1: Repository QR Code

Link: https://github.com/mario-antunes/aiml-society

Part 1: Introduction and Vision

The Challenge in University Management

Current State of Public Services:

- **Siloed Data:** Admissions, Research, HR, and Finance rarely “speak” to each other.
- **Bureaucracy:** Heavy administrative burden for faculty and staff.
- **Student Experience:** Fragmented support services and slow response times.
- **Decision Latency:** Strategic decisions are delayed by manual data processing.

The GenAI Opportunity

From Automation to Augmentation

- **Administrative Efficiency:** Automating routine queries and paperwork.
- **Research Acceleration:** Speeding up literature reviews and grant writing.
- **Personalized Learning:** Adaptive tutoring and curriculum design.
- **Strategic Insight:** Data-driven decision-making using synthesized knowledge.

Part 2: Review of GenAI Paradigms

A Quick Refresher

- **Large Language Models (LLMs):** Predict the next token to generate text, code, and logic (e.g., GPT-4, Claude, Llama).
- **Stable Diffusion (SD):** Generates images/media from text descriptions.
- **Utility:** Excellent at creation, summarization, and translation.
- **Limitation:** Hallucinations and lack of real-time knowledge.

Retrieval-Augmented Generation (RAG)

Grounding the AI

- **Concept:** Retrieve relevant documents from a vector database *before* generating an answer.
- **Process:** User Query -> Search Database -> Inject Context -> LLM Answer.
- **University Use Case:** “What is the policy for sabbatical leave?” (Retrieves the specific PDF from the HR portal).
- **Pros:** Reduces hallucination.
- **Cons:** Limited reasoning; requires perfect retrieval.

Context-Augmented Generation (CAG)

The “Big Memory” Approach

- **Concept:** Instead of retrieving snippets, feed the *entire* relevant dataset into the LLM’s massive context window (e.g., 1M+ tokens).
- **Cache Prefill:** “Caching” the context so it doesn’t need to be re-processed for every query.
- **University Use Case:** Uploading an entire semester’s worth of lecture notes and asking complex synthesis questions.
- **Pros:** Better global reasoning than RAG (connects dots across the whole doc).
- **Cons:** Costly if not cached; high latency for massive contexts.

The Limit of “Passive” GenAI

Why we need more than Chatbots

- **RAG/CAG are Passive:** They wait for a user query and respond. They do not *act*.
- **The Missing Link:**
  - They cannot update the database.
  - They cannot send emails.
  - They cannot navigate complex, multi-step workflows autonomously.
- **Solution: Agentic AI**.

Part 3: Agentic AI - The New Frontier

What is Agentic AI?

Definition

- An AI system capable of autonomous **perception, reasoning, action, and learning** to achieve high-level goals.
- **Core Loop:** Observe -> Think (Plan) -> Act (Use Tool) -> Reflect -> Repeat.
- **Shift:** From “Help me write this email” to “Manage the coordination of the faculty retreat.”

The Architecture of an Agent

1. **The Brain (LLM):** Handles reasoning, planning, and decision-making.
2. **Memory:**
    - *Short-term:* Context of the current task.
    - *Long-term:* Vector DBs or Knowledge Graphs (past experiences).
3. **Tools:** Capabilities (Web Search, Code Interpreter, API calls).
4. **Planning:** Breaking down complex goals into sub-tasks (Chain of Thought).

Definition

- **Multi-Agent Systems (MAS):** Multiple specialized agents working together to solve a complex problem that a single agent cannot handle effectively.
- **Roles:** Instead of one generalist LLM, you have a "Researcher," a "Writer," a "Reviewer," and a "Manager."

Collaboration Patterns

1. **Sequential Handoffs:** Agent A finishes a task and passes output to Agent B (e.g., Draft -> Translate).
2. **Hierarchical (Boss/Worker):** A “Manager” agent breaks down the plan and assigns tasks to “Worker” agents, aggregating their results.
3. **Joint Collaboration:** Agents debate and iterate (e.g., a “Red Teaming” agent challenging a “Policy Drafter” agent).

Use Case - University Grant Proposal

The “Grant Writing Team”

- **Agent A (The Researcher):** Scours academic databases for references and latest state-of-the-art.
- **Agent B (The Strategist):** Analyzes the specific grant requirements and scoring rubric.
- **Agent C (The Writer):** Drafts the content based on A and B.
- **Agent D (The Critic):** Reviews the draft against the rubric and requests revisions from Agent C.
- **Outcome:** High-quality, compliant proposals with minimal human intervention.

Use Case - Student Admission Processing

The “Admissions Committee Support”

- **Agent A (OCR/Parser):** Extracts data from transcripts and PDFs.
- **Agent B (Evaluator):** Checks prerequisites against specific program requirements.
- **Agent C (Flagging):** Identifies anomalies or exceptional achievements for human review.
- **Agent D (Communicator):** Drafts personalized status updates for applicants.
- **Benefit:** Reduces processing time from weeks to hours.

Discussion - Agent Collaboration

Pros & Cons

- **Pros:**
  - **Specialization:** Smaller, cheaper models can be expert at one thing.
  - **Self-Correction:** Agents can check each other's work.
  - **Modularity:** Easy to upgrade one agent without breaking the system.
- **Challenges:**
  - **Loops:** Agents getting stuck in endless debates.
  - **Cost:** Multiple API calls for a single outcome.
  - **Latency:** Slower than a single zero-shot generation.

Connecting to the World

- **The Problem:** Every data source (Student DB, HR System, Library) has a different API. Building agents for each is unscalable.
- **The Solution: Model Context Protocol (MCP)**.
- **Definition:** An open standard that enables AI assistants to connect to systems (content repositories, business tools, development environments) in a uniform way.

How MCP Works

The Ecosystem

- **MCP Host:** The AI application (e.g., Claude Desktop, University Portal AI).
- **MCP Client:** The connector within the host.
- **MCP Server:** A lightweight bridge to specific data (e.g., a "PostgreSQL MCP Server" or "Google Drive MCP Server").
- **Mechanism:** The LLM asks "What tools do I have?" The MCP server replies "You can query the student database." The LLM sends a generic request, MCP translates it.

Use Case - Integrated Campus Management

Scenario: “Book a room for the Ethics 101 makeup class.”

1. **Identity:** Agent uses MCP to verify the requestor’s faculty status via the **LDAP MCP Server**.
2. **Availability:** Agent queries the **Room Scheduling MCP Server** for free slots.
3. **Conflict Check:** Agent checks the **Student Information System MCP Server** to ensure enrolled students don’t have conflicting classes.
4. **Action:** Agent executes the booking.

Discussion - MCP in Public Services

Why it matters

- **Standardization:** Write the connector once, use it with any MCP-compliant LLM (Claude, etc.).
- **Security:** Public services have sensitive data. MCP allows local execution—credentials often stay with the server/host, not the model provider.
- **Interoperability:** Breaks down the “Siloed Data” problem mentioned in Slide 3.

Beyond Vector Similarity

- **The Limitation of Vectors:** RAG retrieves data based on keyword similarity. It struggles with multi-hop reasoning (e.g., “Who is the advisor of the student who failed the prerequisite for the advanced physics course?”).
- **Knowledge Graphs:** Data structured as **Nodes** (Entities) and **Edges** (Relationships).
- **GraphRAG:** Using LLMs to traverse these graphs to find answers.

GraphRAG for Decision Making

Structured Reasoning

- **Mapping Relationships:**
  - *Student -> enrolled in -> Course*
  - *Course -> taught by -> Professor*
  - *Professor -> manages -> Grant*
- **Decision Power:** Allows the AI to understand dependencies and causality, not just text similarity.

Use Case - Curriculum Development

Optimizing Prerequisites

- **Query:** “If we move ‘Intro to Stats’ to Year 2, how does that impact the Engineering major graduation path?”
- **KG Agent Action:**
  1. Trace dependency edges from ‘Intro to Stats’.
  2. Identify all downstream courses (Machine Learning, Advanced Physics).
  3. Identify bottleneck risks.
- **Result:** A decision report highlighting potential graduation delays.

Use Case - Resource & Grant Allocation

Conflict of Interest & Synergy Detection

- **Problem:** Avoiding allocating grants to reviewing committee members or missing interdisciplinary opportunities.
- **KG Agent Action:**
  1. Map all researchers, their publications, and their department affiliations.
  2. Identify hidden links (e.g., Co-authorships 3 years ago).
  3. Suggest reviewers who are truly independent.
  4. Suggest combining two similar small grant requests into one larger interdisciplinary project.

Part 4: Landscape of GenAI for Public Service Tasks

Coding and IT Automation

Tasks:

- Maintaining legacy university systems (COBOL/Java).
- Writing scripts for data migration.
- Automating IT support tickets. **Models:**
- **Commercial: Claude 3.5 Sonnet** (Current SOTA for coding), **GPT-4o**.
- **Open Source: DeepSeek-V3/R1** (High performance/cost ratio), **Qwen 2.5 Coder**.

Scientific Research Support

Tasks:

- Literature Review.
- Hypothesis generation.
- Data cleaning and analysis. **Models & Tools:**
- **Specialized: Perplexity** (Search), **Elicit** (Research workflows), **Consensus**.
- **Models: Gemini Pro** (2M context window for reading 100s of papers at once), **Google NotebookLM** (Audio summaries/podcasts of research).

Text Manipulation (Admin & Legal)

Tasks:

- Summarizing meeting minutes (Faculty Senate).
- Interpreting government regulations/policies.
- Drafting bilingual communications. **Models:**
- **Commercial: GPT-4o** (Versatile), **Claude 3 Opus** (Nuance and high-quality writing).
- **Open Source: Llama 3 (70B/405B)**, **Mistral Large**.

Image Generation & Communication

Tasks:

- Creating visual aids for lectures.
- University marketing materials.
- Architectural rendering for campus planning.

Models:

- **Commercial: Midjourney v6** (Best artistic quality), **DALL-E 3** (Easiest instruction following).
- **Open Source: Flux.1** (Current SOTA open model), **Stable Diffusion 3.5**.

Indexing and Knowledge Management

Tasks:

- Digitizing physical archives.
- Semantic search over university regulations.

Techniques:

- **Embedding Models: OpenAI text-embedding-3**, **Cohere Embed v3** (Multilingual).
- **Vector DBs: Pinecone**, **Milvus**, **Weaviate**.
- **Reranking:** Using a reranker model to improve search relevance before sending to LLM.

Complex Decision Making

Tasks:

- Strategic Budget Planning.
- Crisis Management Simulation.

Models (Reasoning Focus):

- **OpenAI o1 / o3-mini:** Uses “Chain of Thought” reinforcement learning to “think” before answering. Best for math, logic, and complex planning.
- **DeepSeek-R1:** Open-source reasoning model comparable to o1.

Part 5: Strategic Choices and Conclusion

Commercial vs. Open Source

The Public Service Dilemma

Feature	Commercial (OpenAI, Anthropic, Google)	Open Source (Llama, Mistral, DeepSeek)
Data Privacy	Data leaves the premise (Enterprise agreements needed).	Can be hosted locally (On-prem/Private Cloud).
Cost	Pay-per-token (OpEx).	Hardware/Hosting costs (CapEx).
Performance	Generally SOTA (State of the Art).	Closing the gap rapidly; sufficient for 90% of tasks.
Control	“Black box” - reliant on vendor updates.	Full control over fine-tuning and updates.

Recommendation for Universities

A Hybrid Approach

1. **Use Commercial Models** for general non-sensitive tasks (marketing, basic coding assistance) where SOTA performance is required.
2. **Use Open Source (Local) Models** for PII (Personally Identifiable Information), student grades, and HR data to ensure GDPR/Data sovereignty compliance.
3. **Invest in Agentic Frameworks:** Focus on building the *system* (MCP, Knowledge Graphs) rather than just prompting a chatbot.

Conclusion

The Path Forward

- GenAI is shifting from **Chat** to **Work**.
- **Agentic AI** allows public services to scale personalized support and automate complex workflows.
- **Knowledge Graphs** and **MCP** are the infrastructure requirements to make agents reliable and useful.
- **Next Step:** Identify one high-friction administrative workflow in your university and pilot a Multi-Agent system to solve it.

Q&A