1 Introduction

This protocol defines the evaluation process for developing Embedded Multi-Agent Systems (Embedded MAS) solutions adopting a BDI (Belief-Desire-Intention) model and incorporating ethical factors aligned with the European Union's guidelines for trustworthy AI, UNESCO, OECD, and IEEE. Students will work through structured stages, progressively building a system concept while integrating both technical and ethical considerations into their designs. The goal is to stimulate critical thinking about how embedded autonomous solutions should be developed responsibly and intelligently. Each project must aim to produce an embedded, distributed, and ethically responsible autonomous system for one specific domain.

2 Scope and Applicability

Students must choose one of four predefined application domains to serve as the context for designing their Embedded MAS. Each domain presents specific challenges, requiring autonomous and distributed solutions grounded in agent-based reasoning.

2.1 Smart Homes

Description	Design embedded MAS solutions that manage residential environments intelligently. Typical challenges: energy management, security monitoring, appliance control, and resident assistance.
Challenges	Not limited to: • Energy Management; • Security Monitoring; • Appliance Control; • Resident Assistance.
Functions	Examples of agent behaviors: • Monitoring temperature and adjusting HVAC systems; • Detecting unusual activity and notifying residents; • Optimizing energy consumption based on occupancy.

Table 1: The Smart Home Domain.

2.2 Crisis Events Management

Description	Design embedded MAS to support detection, coordination, and response during critical events such as natural disasters or emergencies.
Challenges	Not limited to: • Detection and emergency response during critical events.
Functions	Examples of agent behaviors: • Monitoring environmental risks (e.g., gas leaks, flooding); • Guiding evacuation routes; • Coordinating communication with emergency responders.

Table 2: The Crisis Events Domain.

2.3 Autonomous Vehicles

Description	Design embedded MAS within autonomous vehicles to support real-time decision-making and coordination with other road users or entities in a smart city.
Challenges	Not limited to: • Autonomous Navigation; • Negotiation; • Safe Interaction Among Vehicles.
Functions	Examples of agent behaviors: • Obstacle detection and avoidance; • Lane merging negotiation; • Adaptive route planning with dynamic updates

Table 3: The Autonomous Vehicles Domain.

2.4 Medical Assistance

Description	Design embedded MAS for health monitoring, assisted living, or emergency interventions.
Challenges	Not limited to: • Health Monitoring; • Emergency Support; • Safe Interaction Among Vehicles.
Functions	Examples of agent behaviors: • Monitoring vital signs and recognizing anomalies; • Triggering alarms for emergency medical response; • Managing daily routines of dependent patients.

Table 4: The Medical Assistance Domain.

3 The Ethical Requirements

Students must address 7 EU-inspired principles during system evaluation:

- Sustainable Development and Social Well-Being
- Justice and Non-Discrimination
- Responsibility and Accountability
- Privacy and Data Protection
- Robustness and Security
- Human Oversight
- Transparency and Explainability

4 System Design Stage

This stage guides students through the structured design of their Embedded MAS. It begins with assigning a domain and progresses through defining, evaluating, and modeling functional and technical requirements. Students will define the system's purpose, evaluate feasibility and implementation difficulty, select appropriate hardware and software, and produce visual and architectural models illustrating how agents and devices interact by exchanging messages, perceptions, and actions (via KQML).

Deliverables:

- A complete set of functional and classified requirements
- A mapped list of technologies (hardware/software)
- Three system diagrams
- Device-agent mapping
- A unified architecture diagram
- A hardware sketch using simulation tools

4.1 Select the Domain

In this initial design substep, students must work within a randomly assigned application domain. Each domain represents a real-world scenario where embedded intelligent agents can be deployed to autonomously solve complex, distributed problems.

To promote fairness and diversity of solutions, each student will receive their domain via a random draft process conducted by the professor.

4.2 Identify Functional Requirements

In this step, students must identify and describe the core functional requirements of the Embedded MAS they are designing. Functional requirements define what the system must do — the tasks, services, or operations the system must support to achieve its objectives within the selected domain (e.g., Smart Home, Crisis Management, etc.).

Students must focus on what the system should do, not how it will be implemented at this stage.

Main Goals:

- Define a clear and specific set of functionalities the MAS must perform
- Align system requirements with real-world challenges of the selected domain

Deliverable: Students must fill out Table 5 listing at least 3 to 5 key functional requirements, describing each clearly and concisely.

ID	Description	Comments / Context
FR1		
FR2		
FR3		
FR4		
FR5		
FR6		
FR7		
FR8		

Table 5: Functional Requirements

4.3 Ethical Alignment and Reflection

In this step, students must explicitly reflect on how ethical principles are addressed in their system. Using the ethical guidelines established in the protocol, students will analyze whether their current system satisfies ethical expectations regarding privacy, accountability, non-discrimination, transparency, and others. If gaps are identified, students must revise or define new requirements to address these ethical considerations. This step ensures the project functions technically and adheres to principles of trustworthy AI and responsible system design.

Main Goals:

- Identify how ethical principles are being fulfilled
- Revise or extend the system to meet these principles
- Promote responsible autonomy and accountability in Embedded MAS design

Deliverables:

1. Table 6 linking ethical principles to each functional requirement, with gaps noted
2. A set of revised or new requirements derived from the ethical analysis

Ethical Principle	Justification
Sustainable Development and Social Well-Being	
Justice and Non-Discrimination	
Responsibility and Accountability	
Privacy Preservation and Data Protection	
Robustness and Security	
Human Oversight	
Transparency and Explainability	

Table 6: Ethical Alignment and Reflection

4.4 Classify Requirements

In this step, students must evaluate each functional requirement previously defined by assigning two scores:

- **Difficulty**: how challenging it would be to implement the requirement (1 = easy, 5 = very difficult)
- **Viability**: how feasible it is to implement the requirement with available resources and technologies (1 = low feasibility, 5 = highly feasible)

This classification helps guide system planning by highlighting which requirements are essential, feasible, and worth focusing on early in the design.

Main Goals:

- Assess the complexity and feasibility of each functional requirement
- Guide the prioritization process in the next step

Deliverable: Students must complete Table 7 for each requirement.

4.5 Prioritize Requirements

In this step, students must organize their functional requirements by priority. Each requirement should be labeled as High, Medium, or Low priority, based on how essential it is to the core operation of the system and how practical it is to implement.

Students will determine a priority level (High, Medium, or Low) for each requirement based on the combination of difficulty and viability, along with a brief justification. The priority should reflect a balance between importance to the system's goals and practical constraints like effort, cost, or available technology.

Main Goals:

- Define a clear execution order for system features
- Emphasize requirements critical to the system's functioning
- Facilitate staged and strategic development

Deliverable: Students must assign a priority level (Table 7) to each requirement and justify their decision, building on the difficulty and viability scores from the previous step.

ID	Difficulty (1–5)	Viability (1–5)	Priority	Justification
FR1				
FR2				
FR3				
FR4				
FR5				
FR6				
FR7				
FR8				

Table 7: Requirement Classification

4.6 Identify Hardware and Software Technologies

In this step, students must identify the hardware components and software technologies needed to implement each functional requirement prioritized in the previous step. It specifies sensors, microcontrollers, actuators, communication interfaces, and relevant software frameworks or libraries (e.g., agent platforms, communication middleware, etc.).

Each requirement must be linked to its corresponding technological needs, emphasizing how the system will be operationalized within an Embedded MAS.

Main Goals:

- Define concrete hardware and software for each requirement
- Ensure alignment between intended system functionality and technical feasibility
- Provide a realistic technological foundation for subsequent development and simulation

Deliverable: Students must complete Table 8 linking each requirement to specific hardware and software components, justifying their selections.

ID	Component	Type	Justification
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	
		Hw ☐ Sw ☐	

Table 8: Component Identification and Classification

4.7 Map Devices and MAS Agents

In this step, students must establish a clear relationship between each physical device and the MAS agents that operate on or with it. For each device, students must list its hardware components (e.g., sensors, microcontrollers, actuators), the agents assigned to that device, and any relevant implementation notes or responsibilities.

This mapping ensures that each element of the physical system is explicitly tied to agent-based control, enabling a modular and coherent embedded architecture.

Main Goals:

- Establish traceability between hardware elements and agent logic
- Support modularity and embedded deployment

Deliverable: Students must complete Table 9 and Table 10, linking devices to their associated hardware components, agents, and assigned responsibilities, and learn how different embedded MASs communicate

Device ID	Component	Associated Agent	Perception / Action

Table 9: Device–Agent Mapping: Perception and Action Responsibilities

Device ID	Target Entity	Force	Content

Table 10: The connection and interaction between devices

4.8 Develop System Diagrams

In this step, students must visually represent the structure of their Embedded MAS. Three types of diagrams are required:

- **Hardware System Diagram** – Shows how physical components (sensors, microcontrollers, actuators, communication interfaces, etc.) are interconnected within a device.
- **Internal Agent Interaction Diagram** – Illustrates how agents from one Embedded MAS internally communicate and coordinate with each other, using KQML performatives to represent message types and belief exchange.
- **Inter-MAS Communication Diagram** – Depicts how agents from different Embedded MAS units exchange information or coordinate actions across devices or systems, again using KQML-based communication. This diagram should clarify the types of messages exchanged, the agents involved, and the structure of communication between devices.

These diagrams provide a comprehensive view of how the physical infrastructure, local agent reasoning, and distributed system coordination work together in the proposed design.

Main Goals:

- Visualize the internal and external architecture of the Embedded MAS system
- Clarify both local (intra-device) and distributed (inter-device) communication
- Support validation, refinement, and integration of the overall solution

Deliverable: Students must produce three distinct diagrams, each labeled and explained, ensuring consistency with the functional and technological decisions of the project.

4.9 Provide Full System Architecture

Students must consolidate all previous design decisions into a coherent and complete system architecture in this step. The architecture must describe how devices, agents, communication flows, and hardware components integrate to form a functional Embedded MAS.

The architecture should incorporate both physical and logical elements, including:

- The connection and interaction between devices
- The deployment of agents across devices
- Communication mechanisms (e.g., KQML, sensor data flow)
- Any embedded platforms or middleware used

Students should present the architecture in a structured format such as diagrams, annotated schematics, or layered architecture views.

Main Goals:

- Integrate all components into a unified architectural model
- Represent both hardware and agent-level perspectives
- Enable clear validation of the system's overall design

Deliverable: Students must submit a complete and labeled representation of the system architecture that aligns with previous steps and diagrams.

5 System Development Stage

In this step, students begin the development of their Embedded MAS by implementing a functional prototype. The goal is to validate the architectural design by partially realizing both the hardware and MAS components. Development should follow the decisions documented in the system architecture, focusing on creating a minimal but working system that demonstrates agent perception, interaction, and actuation.

This step includes two complementary tasks:

- Hardware Sketch and Simulation – Using a simulator such as SimulIDE, students must reproduce the proposed hardware layout for each Embedded MAS, ensuring that all relevant components are represented and interconnected according to the system architecture. This sketch serves as a proof-of-concept to test signal flows and constraints.
- MAS Prototype Implementation – A basic implementation of the BDI-based agents and their communication should be created, using performative messages (e.g., via KQML). The prototype must show how agents process perceptions (belief updates), initiate actions, and exchange information.

Main Goals:

- Begin realization of the proposed Embedded MAS
- Test communication between hardware and agents in a simulated environment
- Verify integration of beliefs, actions, and intra-device messaging
- Prepare the system for ethical assessment and revision in the next phase

Deliverables:

1. A simulation file or schematic showing the physical layout using SimulIDE (or similar)
2. A working codebase or agent script demonstrating BDI behavior and agent communication

6 Ethical Assessment and Revision

In this final step, students are required to critically assess the ethical dimension of their project. This includes reflecting on the process of integrating ethical principles into the system and identifying any difficulties, conflicts, or unresolved issues encountered during the specification and implementation of ethical requirements.

Students must examine whether:

- Any ethical principles were hard to apply in practice;
- Technical constraints limited ethical implementation;
- Conflicts emerged between ethical requirements and functional goals.

Main Goals:

- Encourage critical thinking about the feasibility and impact of ethical integration in Embedded MAS;
- Identify areas where further improvement, clarification, or support is needed;
- Document open issues to foster ethical awareness in future iterations or similar systems.

Deliverable: A report indicating:

- Difficulties and problems encountered during ethical specification and development;
- Open Issues;
- Suggestions for improving ethical integration in future projects.