SISTEMAS MULTIAGENTES

ING. OTTO PARRA GONZÁLEZ, PhD

CAPÍTULO 3: METODOLOGÍAS PARA EL DISEÑO DE AGENTES

SISTEMAS MULTIAGENTES

METODOLOGÍAS PARA EL DISEÑO DE AGENTES

Designing multiagent systems (MAS) involves creating systems with multiple autonomous agents that interact with each other to achieve common goals or objectives

INTRODUCTION

- A methodology is a systematic process that specifies aspects of the design process. In the case of software development, it typically involves distinct phases such as planning, analysis, design, and implementation. There are several methodologies for designing multi-agent systems (MAS), such as MaSE, Gaia, and MAS-CommonKADS. These approaches result in a top-down, “functional,” hierarchical design.
- Designing multiagent systems (MAS) involves creating systems with multiple autonomous agents that interact with each other to achieve common goals or objectives. The design process for multiagent systems is complex and requires careful consideration of various factors.

METHODOLOGIES TO DESIGN MAS

AGENT-ORIENTED SOFTWARE ENGINEERING (AOSE)

- AOSE is a methodology specifically tailored for designing multiagent systems. It emphasizes modeling agents as the primary entities and focuses on their interactions and behaviors.
- **Key Concepts:**
  - Agent Modeling: Agents are modeled with explicit states, behaviors, and communication protocols.
  - Interaction Protocols: Define rules for how agents interact with each other.
  - Environment Modeling: Consider the environment in which agents operate and interact

BELIEF-DESIRE-INTENTION (BDI) MODEL

- BDI is a cognitive model that represents agents' mental states: beliefs about the world, desires or goals, and intentions to achieve those goals.
- **Key Concepts:**
    - Beliefs: Represent the agent's perception of the environment.
    - Desires: Define the agent's goals or objectives.
    - Intentions: Specify the agent's plans to achieve its desires

HOLONIC MULTIAGENT SYSTEMS

- Holonic systems view agents as autonomous entities organized into hierarchies or holarchies. Each holon is an autonomous agent that can act independently or as part of a larger system.
- **Key Concepts:**
  - Holarchy: Agents are organized into a hierarchy of holons, with each level having specific responsibilities.
  - Autonomy: Agents at different levels have a degree of autonomy.
  - Cooperation: Holons cooperate to achieve common goals

ORGANIZATIONAL MODELS

- This approach focuses on modeling the social structures and organizations within multiagent systems, defining roles, responsibilities, and communication patterns.
- **Key Concepts:**
  - Roles: Define the functions and responsibilities of agents within the organization.
  - Norms: Specify rules or guidelines governing agent behavior.
  - Communication Structures: Describe how agents communicate and coordinate

EMERGENT BEHAVIOR AND SWARM INTELLIGENCE

- Inspired by natural systems, this methodology focuses on designing systems where complex, adaptive behavior emerges from the interactions of simple agents.
- **Key Concepts:**
  - Local Interactions: Agents follow simple rules governing local interactions.
  - Emergence: Complex global behavior emerges from the collective actions of individual agents.
  - Self-Organization: Systems adapt and organize themselves without centralized control

AGENT-BASED MODELING (ABM)

- ABM involves modeling and simulating the interactions of individual agents to understand and analyze the emergent properties of a system.
- **Key Concepts:**
  - Micro-Level Modeling: Agents are modeled at an individual level.
  - Simulation: Use of computational models to simulate the interactions and behaviors of agents.
  - Analysis of Emergent Properties: Study the system-level properties that emerge from individual agent interactions

MAS DEVELOPMENT FRAMEWORKS

- Frameworks provide reusable structures, libraries, and methodologies for developing multiagent systems.
- **Key Concepts:**
  - Reusability: Utilize pre-built components for common agent functionalities.
  - Scalability: Frameworks should support the development of large-scale multiagent systems.
  - Interoperability: Enable agents developed using different frameworks to interact seamlessly

AGENT-ORIENTED SOFTWARE ENGINEERING (AOSE)

(AOSE) is a specialized approach to software engineering that focuses on the design and development of complex systems comprising multiple autonomous agents.

INTRODUCTION

- Agents in AOSE are computational entities that are capable of autonomous decision-making and can interact with each other and their environment to achieve specific goals. AOSE provides a set of methodologies, concepts, and tools to facilitate the modeling, analysis, and implementation of multiagent systems.
- AOSE provides a structured and systematic approach to developing multiagent systems, allowing for the effective modeling, analysis, and implementation of complex, distributed, and autonomous systems.
- The methodologies and tools associated with AOSE aim to address the unique challenges posed by multiagent environments, making it a valuable paradigm for certain applications, such as distributed control, automation, and decentralized decision-making.

AGENT MODELING

- AOSE emphasizes the modeling of agents as the central entities in a system. Agents are conceptualized with explicit states, behaviors, and communication protocols.
- **Key Components:**
  - *State:* Represents the internal condition or information of an agent.
  - *Behavior:* Describes how an agent responds to internal and external stimuli.
  - *Communication Protocols:* Define rules and conventions for agent interactions

INTERACTION PROTOCOLS

- AOSE focuses on defining interaction protocols that govern how agents communicate and cooperate to achieve common goals.
- **Key Concepts:**
  - *Message Passing:* Agents communicate through messages, and protocols define the structure and content of these messages.
  - *Coordination:* Describes how agents synchronize their actions to achieve shared objectives.
  - *Negotiation:* Involves agents reaching agreements through a process of negotiation

ENVIRONMENT MODELING

- AOSE takes into account the environment in which agents operate. This includes the physical or virtual space where agents exist and the resources available to them.
- **Key Considerations:**
  - *Perception:* Agents gather information about their environment through perception mechanisms.
  - *Action:* Agents can perform actions that affect the environment.
  - *Environment Dynamics:* The environment may change dynamically, influencing agent behavior.

ORGANIZATION MODELING

- AOSE often involves modeling the organizational structure within which agents operate. This includes defining roles, responsibilities, and relationships among agents.
- **Key Components:**
  - *Roles:* Define the functions and responsibilities of individual agents within the organization.
  - *Norms:* Specify rules or guidelines that govern agent behavior.
  - *Coordination Mechanisms:* Describe how agents collaborate and coordinate their actions

METHODOLOGIES

- AOSE provides specific methodologies to guide the development of multiagent systems.
- **Example Methodologies:**
  - *Gaia:* Focuses on organizational structures and the relationships between agents.
  - *Tropos:* Emphasizes goal-oriented modeling and analysis.
  - *MaSE:* Incorporates a social perspective, considering both individual agents and the social context

TOOLS AND AGENTS

- AOSE is supported by various tools and programming languages tailored for agent-based systems.
- **Example Tools:**
  - *Jason:* An agent-oriented programming language based on an extended version of the AgentSpeak language.
  - *JADE (Java Agent Development Framework):* A framework for building multiagent systems in Java.
  - *Prometheus:* A toolset for developing and analyzing multiagent systems using the AUML (Agent UML) notation

CHALLENGES AND CONSIDERATIONS

- Designing and implementing multiagent systems with AOSE poses certain challenges.
- **Challenges:**
  - *Scalability:* Ensuring that the system can handle a large number of interacting agents.
  - *Dynamic Environments:* Adapting agents to changes in the environment.
  - *Interoperability:* Ensuring seamless communication between agents developed using different methodologies or tools

GAIA METHODOLOGY

The Gaia methodology is an Agent-Oriented Software Engineering (AOSE) approach specifically designed for modeling and designing complex systems with multiple interacting agents.

INTRODUCTION

- Gaia focuses on the organizational perspective, emphasizing the relationships and interactions among agents within a system. It provides a structured and systematic way to design multiagent systems, taking inspiration from social organizations and their structures.
- The Gaia methodology provides a structured and systematic approach for designing multiagent systems, with a focus on the organizational aspects of agent interactions. It offers a way to model complex systems by considering the social structures that emerge from the interactions among autonomous agents, making it particularly suitable for applications where organizational coordination and collaboration are crucial.

ORGANIZATIONAL VIEWPOINT

- Gaia places a strong emphasis on the organizational viewpoint, considering agents not just as individual entities but as members of an organization with specific roles, responsibilities, and interactions.
- **Key Concepts:**
    - *Agents:* Individual entities with autonomous capabilities.
    - *Roles:* Define the functions and responsibilities of agents within the organization.
    - *Interaction:* Focus on the patterns of interaction between agents

ORGANIZATIONAL ABSTRACTIONS

- Gaia introduces several abstractions to capture the organizational structure and relationships among agents.
- **Key Abstractions:**
  - *Agent:* Represents an autonomous entity capable of interacting with other agents.
  - *Role:* Defines a set of responsibilities and behaviors that an agent can assume.
  - *Position:* Describes the location of an agent within an organizational structure

SOCIETIES AND SOCIOGRAMS

- Gaia introduces the concept of societies, which represent groups of agents organized to achieve specific goals. Sociograms are used to depict the social structures and relationships within these societies.
- **Key Concepts:**
  - *Society:* A group of agents working together to achieve common objectives.
  - *Sociogram:* A graphical representation of the relationships and interactions among agents in a society

INTERACTION PROTOCOLS

- Gaia defines interaction protocols to guide the communication and coordination between agents. These protocols specify the rules and patterns of interaction that agents must follow.
- **Key Components:**
  - *Communication Protocols:* Define how agents exchange information and coordinate their actions.
  - *Norms:* Specify rules or guidelines that govern agent behavior during interactions

ROLES AND RESPONSIBILITIES

- Gaia places a significant emphasis on clearly defining the roles and responsibilities of agents within the organizational structure.
- **Key Considerations:**
  - *Role Definition:* Clearly specifying the tasks, functions, and behaviors associated with each role.
  - *Responsibility Assignment:* Assigning roles to agents based on their capabilities and expertise

METHODOLOGY STEPS

- The Gaia methodology involves a series of steps to guide the design and development of multiagent systems.
- **Example Steps:**
  - *Identifying Agents and Roles:* Define the agents in the system and the roles they play.
  - *Defining Societies:* Identify groups of agents that form societies with shared goals.
  - *Creating Sociograms:* Develop graphical representations of the social structures and relationships.
  - *Specifying Interaction Protocols:* Define rules for communication and coordination among agents.
  - *Refinement:* Refine the organizational structure based on feedback and analysis

TOOL SUPPORT

- Various tools and modeling languages support the Gaia methodology for designing multiagent systems.
- **Example Tools:**
  - *ADELFE:* A toolset for Agent-Oriented Software Engineering that supports the Gaia methodology.
  - *MaSE:* A multiagent system engineering environment that incorporates Gaia concepts.

GAIA METHODOLOGY: RESEARCH

This section includes some papers that report research related to multi-agent systems and applying the GAIA methodology.

https://ieeexplore.ieee.org/abstract/document/9073811

A Systematic Literature Review in Multi-Agent Systems: Patterns and Trends

Publisher: IEEE

Cite This PDF

Mariana Falco ; Gabriela Robiolo **All Authors**

Abstract:
Multi-Agent Systems became a powerful solution to model and solve problems in complex and dynamic environments. While research in this area grew exponentially before 2009, there is a need to understand the status quo of the field from 2009 to June 2017 in order to comprehend the general evolution. The results of a SLR related to Multi-Agent Systems, its applications and research gaps, following Kitchenham and Wholin guidelines are presented in this paper. From the analysis of279 papers (out of3522 candidates), our findings suggest that: a) there is a general decreasing trend of publications (but it is increasing for specific domains), b) only 15% of the papers portrayed a real case study, c) the top 20 were formed by 67 authors, d) the papers were mostly published in journals and conferences, e) there is no unified methodology or framework, f) the top 3 application domains were transport/traffic, healthcare/biology, and logistics/manufacturing, g) MAS interact with different disciplines like machine learning. Finally, the MAS community should work together to close the gaps and unify the field, bridging with other disciplines and industry.

Published in: 2019 XLV Latin American Computing Conference (CLEI)

Date of Conference: 30 September 2019 - 04 October 2019 **INSPEC Accession Number:** 19556168

Date Added to IEEE Xplore: 23 April 2020 **DOI:** 10.1109/CLEI47609.2019.235098

Open Access

Full Length Article

ORIGINAL ARTICLE

Formal Modeling of a Mail Transport System based on Multi-Agent System-of-Systems

Nadeem Akhtar[1], Shafiq Hussain[2]

[1]Assistant Professor, Department of Computer Science and IT, The Islamia University of Bahawalpur, Pakistan
[2]Assistant Professor/Head, Department of Computer Science and IT, University of Sahiwal, Pakistan

ABSTRACT

The role of multi-agent System-of-Systems (SoS) has become important in modern complex systems. SoS is a composition of systems having constituent elements as independent functional autonomous systems. It is a specialized system with greater complexity having an emergent behavior. We have proposed a methodology centered on formal modeling, model checking, and formal verification of a Mail Transport System based on multi-agent SoS. The proposed multi-agent SoS is a safety-critical system that must be correct, safe, and reliable. It must ensure the safety and liveness properties of correctness. Our objective in this work is to propose a formal methodology that ensures correctness properties of safety and liveness of the Mail Transport System. Our contribution consists of specifying Gaia based formal multi-agent requirement and design specifications, the liveness properties are specified using regular expression and the safety property is specified in first-order predicate calculus. The verification of Gaia safety and liveness properties and Gaia organizational abstractions by Finite State Processes (FSP) and Labelled Transition System (LTS). Then these FSP specifications are modeled in Event-B for creating exhaustive proofs.

Keywords: Safety-Critical System, Multi-agent System, System-of-System (SoS); Labelled Transition System (LTS); Correctness; Safety; Liveness; Finite System Processing (FSP)

Author`s Contribution
[1,2]Manuscript writing, Data analysis, interpretation, Conception, synthesis, planning of research, Interpretation and discussion, Data Collection

Address of Correspondence
Nadeem Akhtar
nadeem.akhtar@iub.edu.pk

Article info.
Received: May 26, 2018
Accepted: June 21, 2019
Published: June 30, 2019

Cite this article: *Akhtar N, Hussain S. Formal Modelling of a Mail Transport System based on Multi-Agent System of Systems. J. Inf. commun. technol. robot. appl.2019; 10(1):68-79*

Funding Source: *Nil*
Conflict of Interest: *Nil*

https://www.clei.org/cleiej/index.php/cleiej/article/view/456/390

Tendencies in Multi-Agent Systems: A Systematic Literature Review

CLEI ELECTRONIC JOURNAL, VOLUME 23, NUMBER 1, PAPER 1, APRIL 2019

Tendencies in Multi-Agent Systems: A Systematic Literature Review

Mariana Falco
Universidad Austral, Facultad de Ingeniería, LIDTUA/CONICET
Pilar, Buenos Aires, Argentina, 1623
mfalco@austral.edu.ar

and

Gabriela Robiolo
Universidad Austral, Facultad de Ingeniería, LIDTUA
Pilar, Buenos Aires, Argentina, 1623
grobiolo@austral.edu.ar

Abstract

The application of Artificial Intelligence mechanisms allows the development of systems capable to solve very complex engineering problems. Multi-agent systems (MAS) are one paradigm that allows an alternative way to design distributed control systems. While research in this area grew exponentially before 2009, there is a need to understand the status quo of the field from 2009 to June 2017. An extension of the results of a SLR related to Multi-Agent Systems, its applications and research gaps, following Kitchenham and Wholin guidelines are presented in this paper. From the analysis of 279 papers (out of 3522 candidates), our findings suggest that: a) there were 20 gaps related to agent-oriented methodologies; coordination, cooperation and negotiation; modelling, developing, testing and debugging; b) 24 gaps related to specific domains (recycling, dynamic evacuation, hazard management, health-care, industry, logistics and manufacturing, machine learning, ambient assisted living); and 14 gaps related to specific areas within MAS (A-Teams, dynamic MAS and mobile agents, ABMS, evolutionary MAS, and self-organizing MAS). These gaps specify lines of research where the MAS community must work to achieve the unification of the agent-oriented paradigm; as well as strengthen ties with the industry.

Keywords: Systematic Literature Review, Multi-Agent Systems, MAS Research Gaps, AOSE components.

https://link.springer.com/chapter/10.1007/978-3-031-08202-3_8

Requirements Engineering Processes for Multi-agent Systems

Iderli Pereira de Souza Filho, Giovane D.'Avila Mendonça, Willian Samuel Gerstberger & Gilleanes Thorwald Araujo Guedes

Chapter | First Online: 04 September 2022

Part of the Artificial Intelligence-Enhanced Software and Systems Engineering book series (AISSE,volume 2)

Abstract

Multiagent systems have been proven to be good alternatives to address complex systems, since the complexity can be divided and attributed to many agents specialized in a given facet of the problem. However, developing this kind of system also proved to be complex and generated new challenges to software engineering, which led to the rise of AOSE—Agent-Oriented Software Engineering, an area that aims to create modeling languages and development processes specifically for multiagent systems. Obviously the development of such processes must include the requirements engineering—an area of software engineering dedicated to propose techniques and processes for eliciting, analyzing, specifying, and validating requirements, which is absolutely necessary to develop software that satisfies the needs of their users. Taking this in consideration, in this chapter we describe requirements engineering techniques and processes developed specifically for multiagent systems, highlighting its strengths and weaknesses with emphasis on the coverage of the Belief-Desire-Intention (BDI) model which is used to build cognitive agents.