Mechanisms for Environments in Multi-Agent Systems

ALEX BAINTER - JAY MICHELS - NICK HARLOFF

Acknowledgement

Platon, E., M. Mamei, N. Sabouret, S. Honiden, and H. V. D. Parunak (2007). Mechanisms for Environments in Multiagent Systems: Survey and Opportunities, Autonomous Agents and Multiagent Systems, 14(1):31-47

Problem

- Environments are an explicit and exploitable element of multi-agent systems
  - However, current applications often regard it as an implicit part of the system
- We can assign responsibilities to the environment that would be more difficult to design if we only considered the agents
- Environments can be designed as part of a multi-agent system through **mechanisms**
  - A mechanism is a technical approach to solve a particular problem in the design and development of the environment responsibilities

Purpose

1. Analyze mechanisms used in the MAS research community
2. Survey mechanisms for designing environments
3. Determine research directions and application opportunities

“How can we use mechanisms to exploit the environment in multi-agent systems?”

Mechanisms

NICK HARLOFF

Mechanisms Definition

- Mechanisms
  - Technical approach to address a particular responsibility of the environment

- Two kinds of activities:
  - Interaction Mediation
  - Resources and Context Management

- Boundaries between the activities aren’t strict

Mechanisms Definition Cont'd.

- Mechanism Characteristics
  - Constituent atoms
  - Creation and maintenance
  - Usage

- Moving forward -> mechanism classes in each activity
  - Start with Interaction Mediation (IM)
  - Move on to Resources and Context Management (RCM)

(IM) Environment-Mediated Interaction Channels

- Support discovery and interaction into single mechanism

Name: Environment-mediated interaction channels.
Motivations: Need to create flexible interaction channels to connect agents on the basis of runtime system situation.
Constituent atoms: Message and subscription repositories.
Creation and maintenance: Protocols to let agents join multicast groups, connect to tuple spaces, place subscriptions, or similar.
Usage: Agents exploit interaction channels to easily send and receive messages.
Examples: Multicast, Tuple- and Event-based interaction [2, 8, 11, 17, 21, 23]

(IM) Synchronization Mechanisms

Name: Centralized synchronization mechanisms.
Motivations: Need for supporting the simultaneity of actions for system consistency.
Constituent atoms: Tables storing the locks in the system and holding the resulting effect of simultaneous actions.
Creation and maintenance: Protocols to let new agents access the service and to define coordination policies.
Usage: Coordination and synchronization of agent activities.
Examples: Influence-Reaction Model [20]

Name: Decentralized synchronization mechanisms.
Motivations: Need for supporting the simultaneity of actions for system consistency in distributed settings.
Constituent atoms: Tables of synchronization locks between agents of each region.
Creation and maintenance: Decentralized synchronization algorithm.
Usage: Agents access the tables to coordinate and synchronize.
Examples: Regional Synchronization [58]

(IM) Overlay Networks

- Distributed data structures providing agents views of their network

Name: Overlay networks.
Motivations: To represent and allow the maintenance of agent relationships.
Constituent atoms: Tables representing the physical, interactive, or social surrounding of agents.
Creation and maintenance: Protocols to let new agents join and leave the topology, and to deal with reconfigurations.
Usage: Agents access the tables to interact with each other efficiently.
Examples: Distributed Hash Tables [45, 46, 48], Social Dependency Nets [50]

(RCM) Resources and Context Manager

- Control the access of agents to resources and contextual data
- Serve to communicate context-data and exchange information among agents

Name: Resource and context manager.
Motivations: Need to represent context information and resources in an efficient way.
Constituent atoms: Handling primitives and repositories.
Creation and maintenance: Protocols to wrap new resources types (interfacing and deployment); algorithms to manage repositories.
Usage: Agents access the repositories to interact efficiently.
Examples: TuCSoN, Event Heap [7, 29, 38]

(RCM) Notification of Contextual Events

Name: Notification of contextual events.
Motivations: Need for the production and delivery of event notifications to create dynamic agent contexts.
Constituent atoms: Event dispatcher repositories.
Creation and maintenance: Protocols to let new agents subscribe to event sources and be notified upon event happening.
Usage: Agents trigger reaction on the basis of the events received.
Examples: Interaction filters [2], LoudVoice [6], Tag interactions [43]

(RCM) Overlay Data Structures

- Two types of overlay data structures
  - Pheromones
  - Fields

Name: Overlay data structures.
Motivations: Need for efficient, expressive contextual information.
Constituent atoms: Multiplicity of data spaces to store the overlay data.
Creation and maintenance: Protocols to deploy overlay data structure, maintain their intended distribution, and maintain data consistency.
Usage: Agents access the overlay data structure to get contextual information.
Examples: TOTA [34], UAVs [41], Swarm Linda [36]

Applications

JAY MICHELS

Applying Mechanisms to a Problem

- Mechanisms are usually not exploited individually
  - Combined to tailor a solution similar to an OOP approach w/ design patterns
- Applicable to new problems, as well as “legacy” applications
  - Legacy problems have a mapping between requirements and mechanism combinations
- So… how are these mechanisms combined and exploited?

Types of Applications

- **Simulations** reproduce characteristics of the real-world
  - Uncoupled from the real-world dynamics
  - Multi-Agent based simulations are a good example
- **Pervasive applications** complete the real-world with a computational counterpart that evolves in synchrony
  - Coupled tightly with the real-world dynamics via sensor networks
  - Quickly expanding with advances in sensor technologies
- **Virtual Societies** stand in-between simulations and pervasive applications
  - Inspired by the real-world and reproduce some characteristics
  - Offer support for human activities such as accounting, library management, etc.

Recurring Application Properties

- Design Abstraction
  - Help hide the complexity of underlying details from agents.
  - Help reduce agent complexity by giving the environment explicit assignments.
- Coordination
  - Mechanisms help with complex coordination of agent behaviors
- Separation of Concerns
  - Relates to the idea that the environment cross-cuts MAS.

Interaction Mediation Applications

Type: Simulation

- Experiment
  - Behavior of ants and termites
- Mechanisms used:
  - Overlay network – provides agents various interaction means
  - Environment-mediated interaction channels – complex social behaviors
  - Resource and context manager – when exploiting environmental resources
- Benefits
  - Coordination of agents in a decentralized way
  - Separation of coordination concern from other design issues

Interaction Mediation Applications

Type: Pervasive Application

- Experiments
  - Sensor networks like those in the Agilla agent framework
  - Industry of automated transportation systems and automatic guided vehicles (AGVs)
- Mechanisms used:
  - Overlay network – provides agents various interaction means
  - Environment-mediated interaction channels
  - Decentralized synchronization – when exploiting environmental resources
- Benefits
  - Decentralized synchronization leads to coordination in AGVs
  - Separation of concerns to help agents focus on their functional requirements

Resource/Context Management Applications

Type: Simulation

- Experiment
  - Wide variety of Multi-Agent Based Simulations that rely on ‘ticks’
- Mechanisms used:
  - Notification of context events – think about ‘ticks’ in a MABS
  - Synchronization mechanism – ensure agents actions execute consistently
  - Overlay data structure – manage access rights on agent contexts
- Benefits
  - Separation of concerns by clearly distinguishing resources from agents
  - Design abstraction by keeping time management separate

Resource/Context Management Applications

Type: Virtual Societies

- Experiment
  - An agent-based web-site with recommendation systems
- Mechanisms used:
  - Overlay data structures
  - Resource and context manager
  - Notification of context events
- Benefits
  - Coordination of agents in a decentralized way
  - Separation of coordination concern from other design issues

Mechanism Applications Wrap-Up

- Types of applications
  - Simulation, pervasive applications, and virtual societies
- Beneficial properties of applying mechanisms
  - Design abstraction, coordination, and separation of concerns
- Wide variety of ways to combine and exploit mechanisms
  - Using a variety of mechanisms typically yield the best design and result

Further Research

ALEX BAINTER

Research Directions and Opportunities

- Exploitable Domains
  - Normative Systems
  - Embodied Conversational Agents

Research Directions and Opportunities

- Mechanisms allow us to transition between concepts to engineering
- Mechanisms can be used as incentives to exploit the environment by providing abstractions

New research directions:

1. Continuation of surveying current mechanisms
2. New mechanisms for under-researched areas
3. High-level mechanisms

Conclusions

- Environments are under-utilized, but can be assigned responsibilities through mechanisms

Two consequences of mechanisms identification:

1. Reveals design idioms that are reusable and composable
2. Reveals unexplored combinations of mechanisms which require further research

Thanks for listening…

Time for questions!