Towards Automated Development of Multi-Agent Systems Using RADE

Xiaoqin (Shelley) Zhang & Haiping Xu
Department of
Computer and Information Science
University of Massachusetts Dartmouth

June 26, 2006

Motivation

- Multi-Agent System - Multiple autonomous/semi-autonomous entities (agents) interact with each other
- A suitable programming paradigm for distributed information systems and applications
- It is not easy to
  - Develop MAS
  - Test and Maintain MAS
  - Reuse current system for a different domain

Separate Concerns
Three-layered development process

AIPI Model
Application Independent Platform Independent Model

Defitintion of Role, Role Organization, Role Space, Agent, Agent Society

⇩

ASPI Model
Application Specific Platform Independent Model

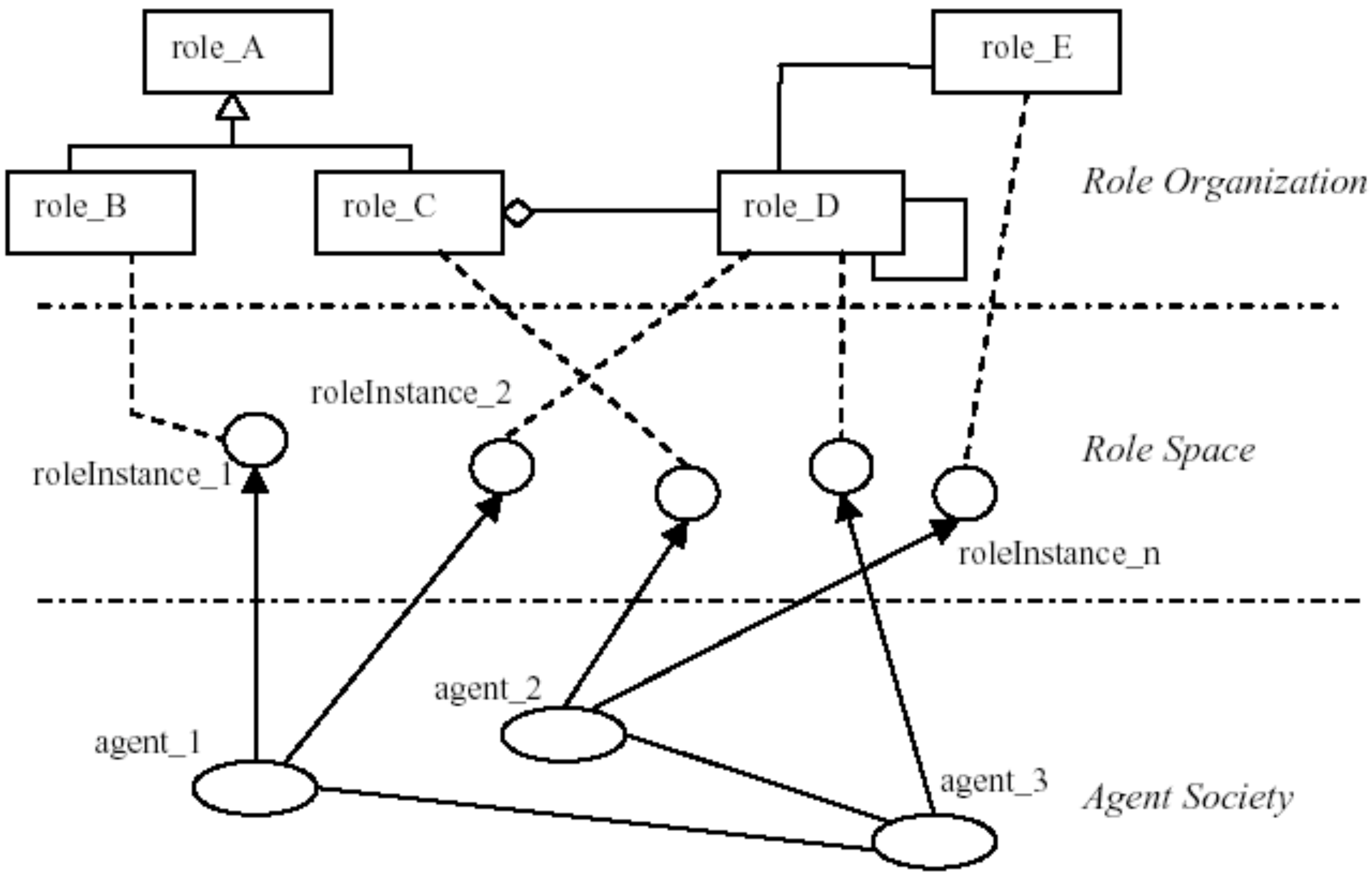
Defitintion of specific role classes, role organization class, agent class, agent society class, and A-R maping

⇩

ASPS Model
Application Specific Platform Specific Model

Defitintion of software platform, middleware and communication mechanism

Role-Based Agent Development Environment (RADE)



Agent Definition

Agent

$attributes : \mathbb{P}\, Attribute$

$motivations : \mathbb{P}\, Motivation$

$utilityFunction : MQState \rightarrow utility$

$sensor : Environment \nrightarrow SensorData$

$reasoningMechanisms :$

$\mathbb{P}\, SensorData \times \mathbb{P}\, Motivation \rightarrow \mathbb{P} \downarrow Role$

$\mathbb{P}\, SensorData \times \mathbb{P}\, Motivation \times \mathbb{P} \downarrow Role$

$\rightarrow \mathbb{P}\, CurrentGoal$

$\mathbb{P}\, SensorData \times \mathbb{P}\, Motivation \times \mathbb{P}\, CurrentGoal$

$\rightarrow \mathbb{P}\, CurrentSchedule$

$executionMechanisms :$

$\mathbb{P}\, SensorData \times \mathbb{P}\, CurrentPlan \rightarrow newEnvironment$

$rolesTaken : \mathbb{P} \downarrow Role$

Agent Motivation

- Agent can devote to multiple tasks that relates to different goals.
- Motivation
  - any desire or preference that can lead to the generation and adoption of goals
  - affects how the agent satisfy those goals
- Motivation Quantities (MQs)
  - Each MQ is associated with a preference function
  - Each agent has a set of MQs it tracks and collects

$$U_{fi}(MQ_i \to U_i)$$

- Motivation of a Personal Assistant Agent

$$U_{agent} = \gamma(U_i, U_j, U_k, \ldots)$$

  - MQmanageAddressBook, MQorganizeActivities, MQpurchaseItems

MQ Extension

- Original MQ Framework (Wagner & Lesser 01)
  - Assume all MQ types are designed by the user when the agent is created
  - The types of MQ are fixed in the runtime of the system
- Introduce subject of MQ

$MQ_{organizeActivitiesForUserA}$ $MQ_{organizeActivitiesForUserB}$

$MQ_{name}(MQ_{subject})$

MQ
name : String
subject : $\mathbb{P}$ entity

$MQ_{organizeActivities}(A)$ and $MQ_{organizeActivities}(B)$ $MQ_{organizeActivities}(User)$

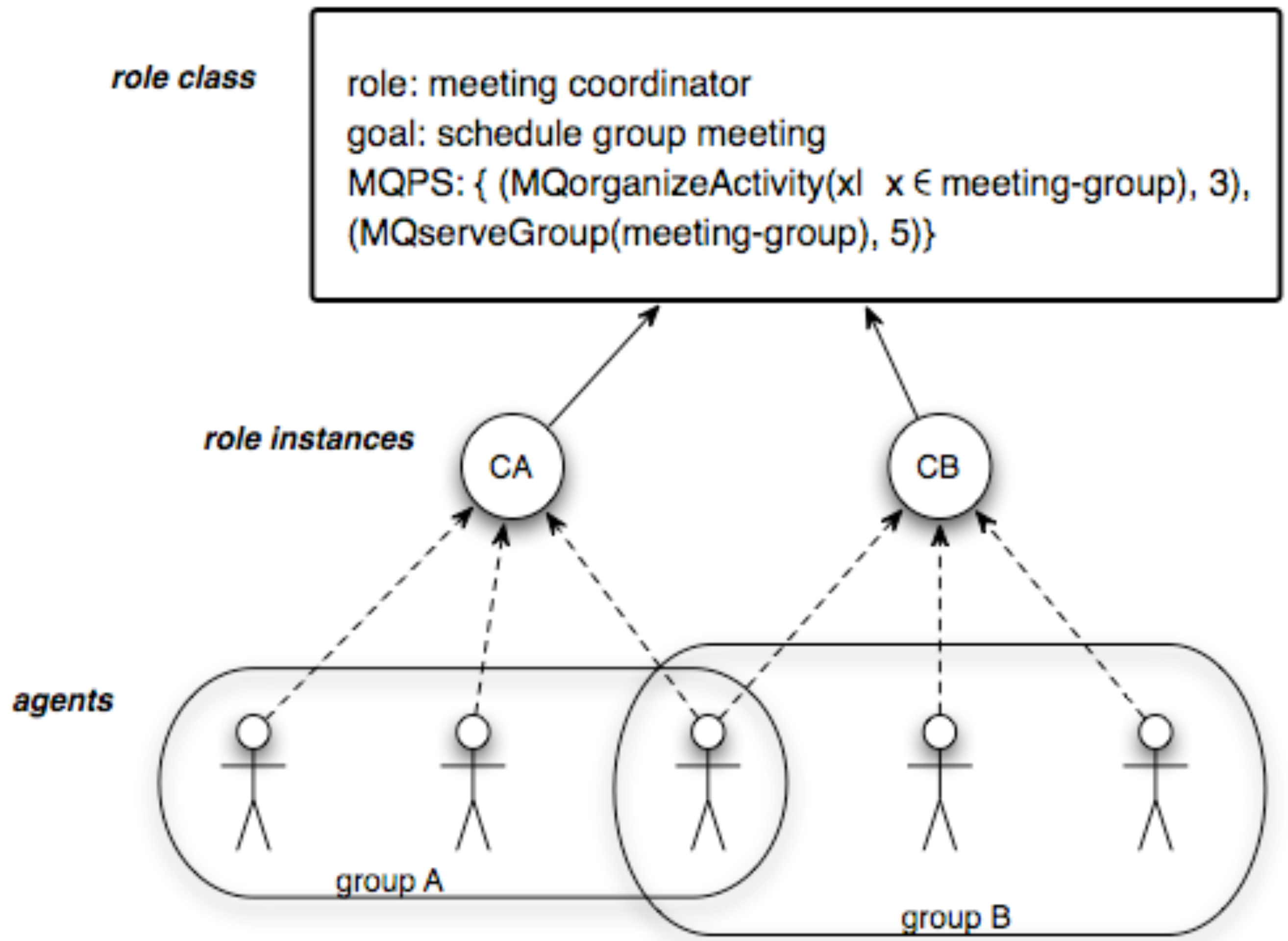
Subject of MQ

- Definition of MQ Subject
  - List a set of entities by enumeration
    - id1, id2, …idn.
  - Specify the conditions for an entity to belong to this set $\{x \mid x \in group_A\}$
- MQ types MQi and MQj are identical iff:
  - Name(MQi) == Name(MQj)
  - Subject(MQi) $\supseteq$ Subject(MQj) && Subject(MQj) $\supseteq$ Subject(MQi)
- MQ types MQi is a special case MQj (MQi $\subset$MQj) iff:
  - Name(MQi) == Name(MQj)
  - Subject(MQi) $\subset$ Subject(MQj)

Role Definition

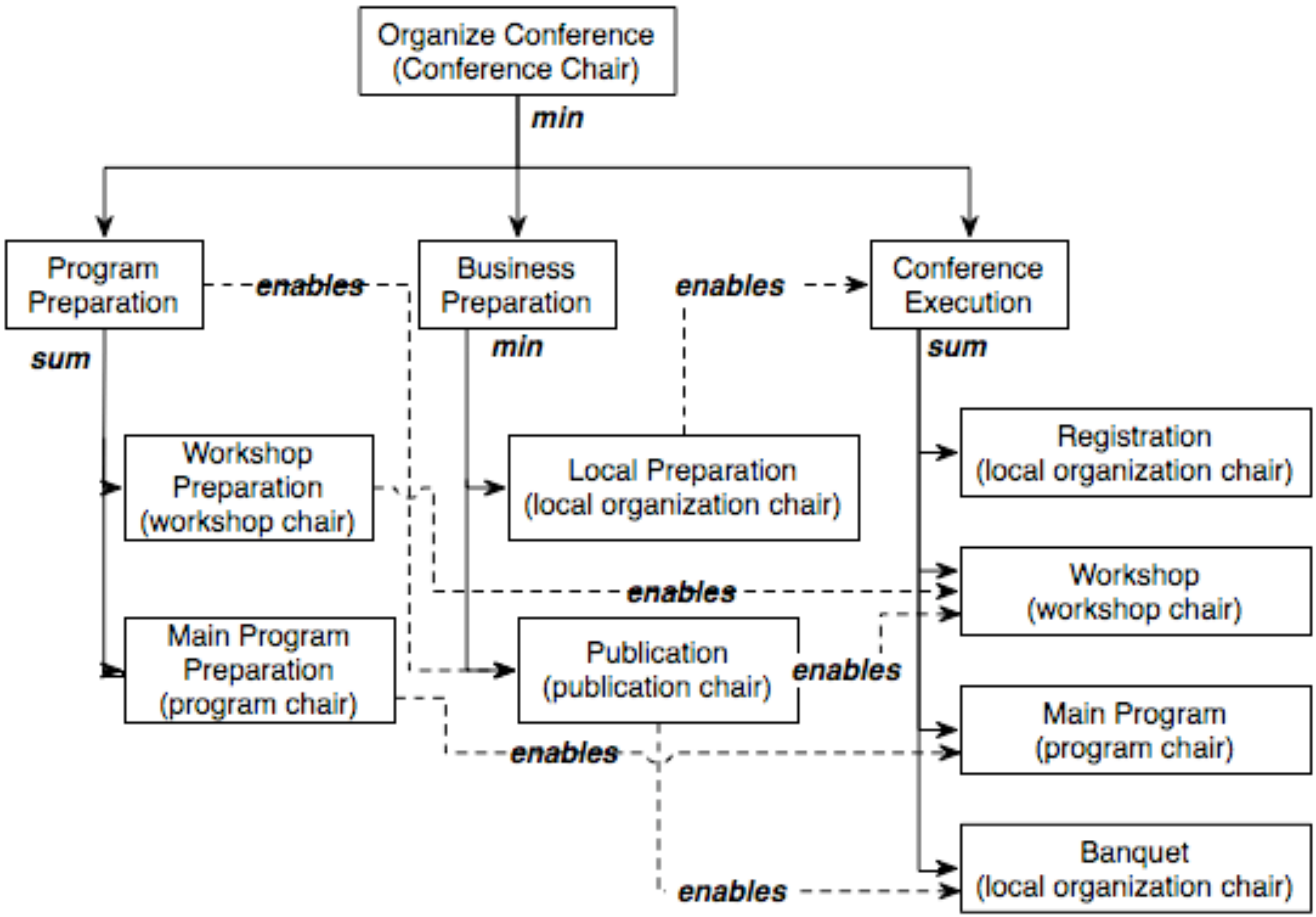
- A Set of Goals, each goal is defined by:
  - Goal name
  - A MQ Production Set $MQPS = \{(MQ_i, q_i), (MQ_j, q_j), (MQ_k, q_k)\ldots\},$

Meeting Coordinator Role Example



Role Definition

- A Set of Goals, each goal is defined by:
  - Goal name
  - A MQ Production Set $MQPS = \{(MQ_i, q_i), (MQ_j, q_j), (MQ_k, q_k)\ldots\},$
  - Each goal is associated with a plan tree
    - a hierarchal description of the alternatives to accomplish a goal
    - R-TAEMS (Role-Based Task Analyzing, environment Modeling, and Simulation language), extending TAEMS (Decker & Lesser).



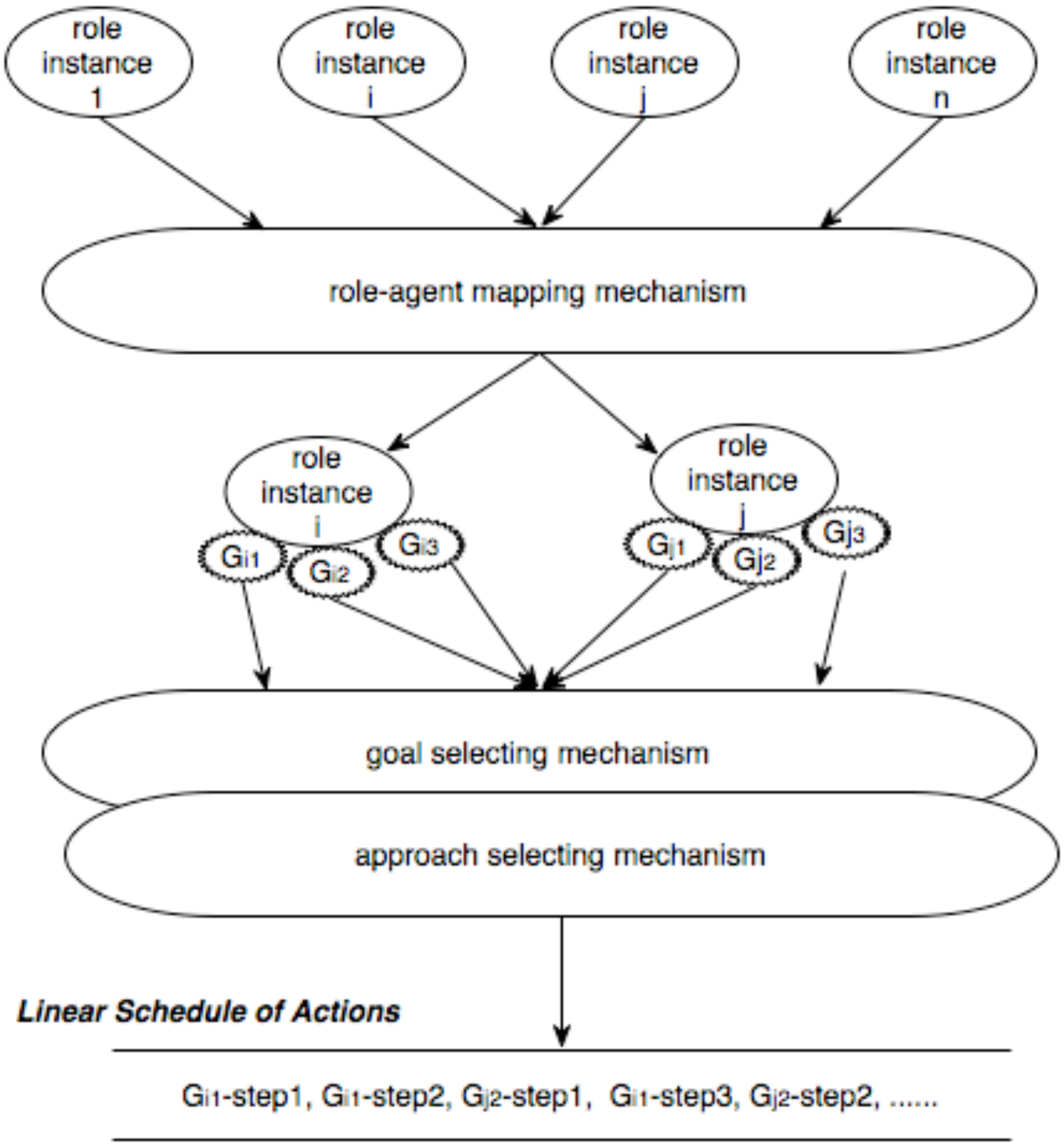
Organize Conference
(Conference Chair)
min
Program
Preparation
enables
Business
Preparation
enables
Conference
Execution
sum
min
sum
Workshop
Preparation
(workshop chair)
Local Preparation
(local organization chair)
Registration
(local organization chair)
Workshop
(workshop chair)
enables
Main Program
Preparation
(program chair)
Publication
(publication chair)
enables
Main Program
(program chair)
enables
Banquet
(local organization chair)
enables

Mapping from Role Instance to Agent

- How does an agent select the role instances it wants to take?
  - An agent is interested in a role instance if some of the goals belong to the role instance match the agent's motivation.
- How to verify the qualification of an agent for a role instance?
  - The verification process is executed by the creator of the role instance.
    - Whether the agent (A) has the capability to take this role instance (R).
    - Whether this role instance is consist with other role instances the agent currently has, based on the incompatibility relationships defined in the role organization.

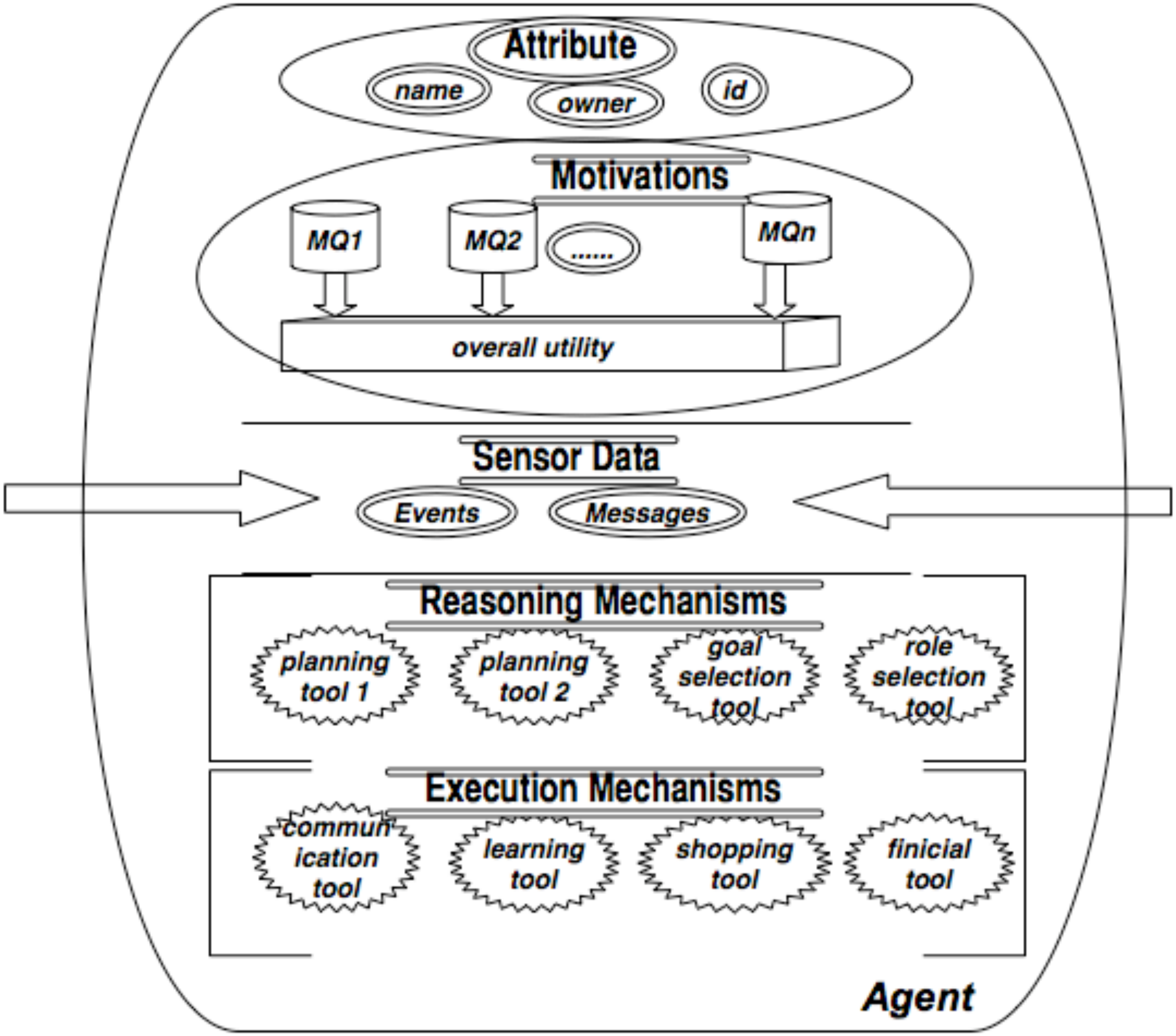
Reasoning Mechanisms

- Decide what roles the agent should take or release given the agent's:
  - motivation
  - current roles it is taking
  - the resource and time constraints.
- Decide what goals the agent should pursue
  - the agent may take multiple roles
  - each role may have multiple goals
- Decide how to achieve a goal given:
  - the available alternative
  - resources and time constraints.
  - some planning and scheduling mechanisms are needed for this decision.


role instance 1
role instance i
role instance j
role instance n
role-agent mapping mechanism
role instance i
role instance j
Gi1
Gi2
Gi3
Gj1
Gj2
Gj3
goal selecting mechanism
approach selecting mechanism
Linear Schedule of Actions
Gi1-step1, Gi1-step2, Gj2-step1, Gi1-step3, Gj2-step2, ...

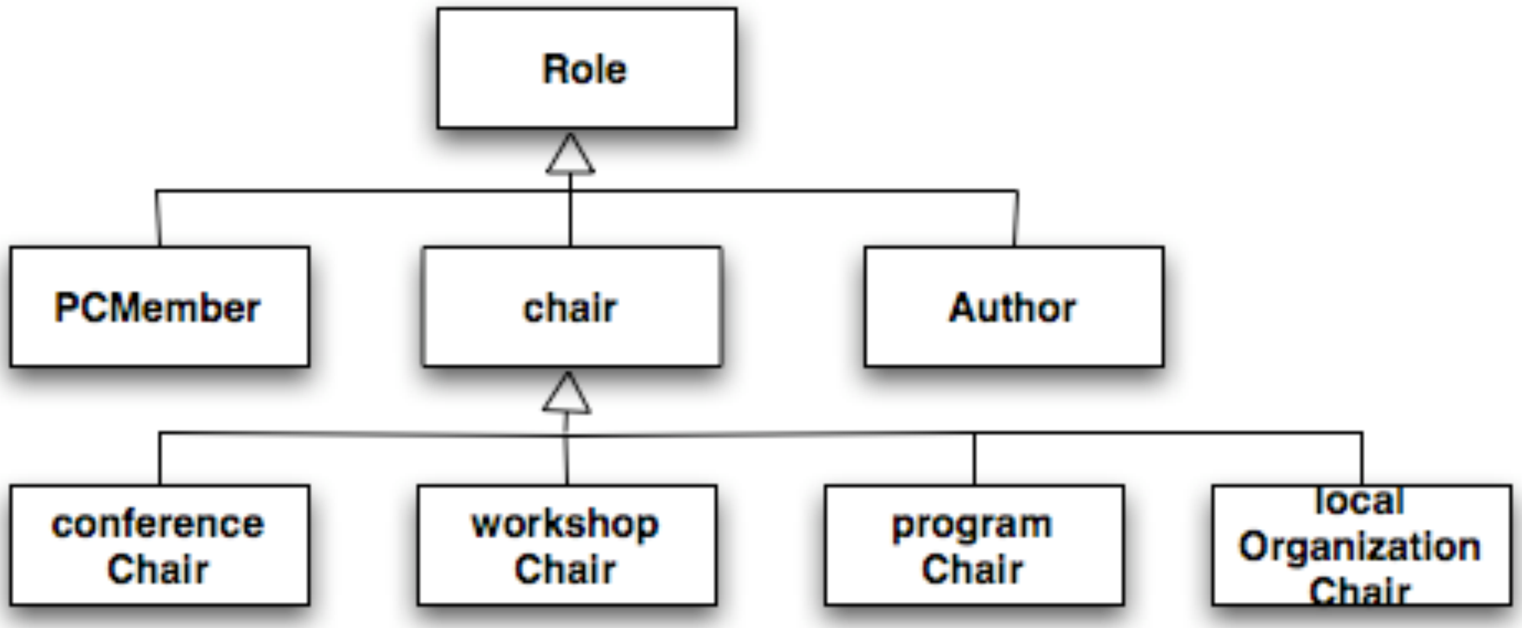
Execution Mechanisms

- Generate the output, which changes the environment
- Robot agents - their actors, such as motors
- Software agents - the primitive actions
  - Some are domain-dependent.
    - the personal assistant agent is build with execution mechanism to perform an online purchase
  - Other are application-independent but platform-dependent
    - such as sending a message.
  - Some common execution mechanisms can be built as toolkits and reused for different applications.



Attribute
name
owner
id
Motivations
MQ1
MQ2
......
MQn
overall utility
Sensor Data
Events
Messages
Reasoning Mechanisms
planning tool 1
planning tool 2
goal selection tool
role selection tool
Execution Mechanisms
commun ication tool
learning tool
shopping tool
finicial tool
Agent

Case Study: Conference Organization



ChairRole

$permissions : \{createNewRoleInstances\}$
$protocols : \{coordinationAsLeader\}$

ConferenceChairRole

$goals : \{organizeConference,$
$MQPS = \{(MQ_{professionalService}, 10)\}\}$
$planTrees : \{RT\text{ÆMS } specification\}$

AuthorRole

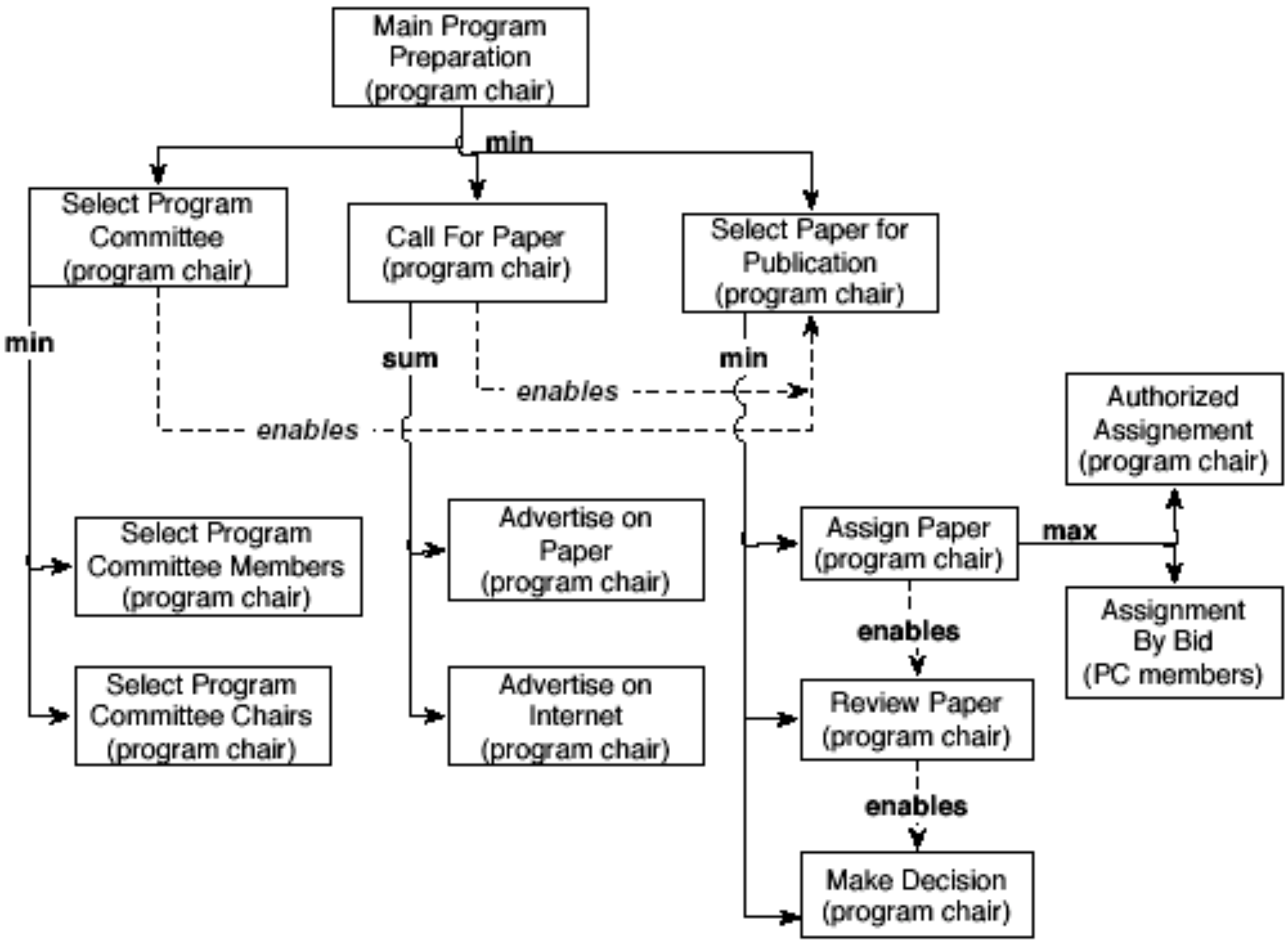
$goals : \{publishPaper,$
$MQPS = \{(MQ_{researchAccomplishment}, 5)\}\}$
$planTrees : \{RT\text{ÆMS } specification\}$

PCMemberRole

$goals : \{reviewPaper,$
$MQPS = \{(MQ_{professionalService}, 1)\}\}$
$planTrees : \{RT\text{ÆMS } specification\}$

ProfessionalAgent

$name : String$
$motivations : \{MQ_{researchAccomplishment}, MQ_{professionalService}\}$
$rolesTaken : \mathbb{P} \downarrow Role$
$reasoningMechanisms : \{planning, scheduling, roleSelection\}$
$executionMechanisms : \{communication, coordination\}$


Main Program Preparation (program chair)
min
Select Program Committee (program chair)
Call For Paper (program chair)
Select Paper for Publication (program chair)
min
sum
min
enables
enables
Authorized Assignement (program chair)
Select Program Committee Members (program chair)
Advertise on Paper (program chair)
Assign Paper (program chair)
max
Assignment By Bid (PC members)
enables
Select Program Committee Chairs (program chair)
Advertise on Internet (program chair)
Review Paper (program chair)
enables
Make Decision (program chair)

Conclusion and Future Work

- A general design of agent architecture for RADE framework.
  - Define the agent's motivation based on the extension of MQ framework
  - Define the goal with MQ production set
  - Develop RTAEMS language to represent the plan trees
  - Describe the role-agent mapping mechanisms and criteria.
- Future work
  - Implement an extended RADE framework including agent design
  - A set of plug-in toolkits for agent reasoning, execution and collaboration
  - A demo of an automated generated multi-agent system and its operation on one application domain.