Multi-agent Semantic Web Systems: Agent Coordination

Ewan Klein

MASWS — 4 March 2008
1 Introduction

2 Electronic Institutions

3 Lightweight Coordination Calculus

4 Summary
Recap

- Centralized vs. non-centralized architecture
- Coordination via middle-agents
- Communication conventions:
  - content
  - performative verbs
  - dialogues, message sequencing
- Electronic Institutions:
  - inspired by human organizations
  - provides framework to govern interaction
More on Electronic Institutions (EI)
Today

- More on Electronic Institutions (EI)
- Lightweight Coordination Calculus (LCC)
# Multi-Agents Systems (MAS)

<table>
<thead>
<tr>
<th>Agent Technologies</th>
<th>WS Counterpart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Protocols (EI, LCC)</td>
<td>OWL-S</td>
</tr>
<tr>
<td>Agent Ontologies</td>
<td>RDF(S), OWL</td>
</tr>
<tr>
<td>Agent Communication (KQML, FIPA-ACL)</td>
<td>WSDL, SOAP</td>
</tr>
<tr>
<td>Agents</td>
<td>Web Services</td>
</tr>
</tbody>
</table>
Conversations are grouped into **scenes**.
Electronic Institutions Recap

- Conversations are grouped into **scenes**.
- Agents participate in scenes by virtue of a role — can play different roles in different scenes.
Electronic Institutions Recap

- Conversations are grouped into scenes.
- Agents participate in scenes by virtue of a role — can play different roles in different scenes.
- Scenes are connected into a performative structure; governs how agents can move from one scene to another.
Conversations are grouped into scenes.

Agents participate in scenes by virtue of a role — can play different roles in different scenes.

Scenes are connected into a performative structure; governs how agents can move from one scene to another.

Participation in an EI circumscribes and constrains interactions with potentially open set of agents / WS.
Electronic Institution

Yellow boxes: scenes
Blue arrows: performative structure
Grey box: scope of institution
Scene Negotiation

- FSM for negotiation scene.
- Like the script of a play;
- each actor has a copy.
- Agents can leave scene at any point.
LCC Overview

- **LCC designed as a refinement of Electronic Institutions:**
  - Includes notions of scene and role;
  - ‘protocol’ corresponds roughly to institution.
  - Scene cannot begin until all agents are present;
  - agents maintain a given role throughout the scene;
  - agents cannot leave until scene is complete.

- **EI requires work to go from an abstract specification to an agent-based implementation.**

- **LCC is intended to be an executable specification.**
Simplified Example: Protocol

- Following exposition in Walton Ch. 6 — notation a bit different from recent LCC papers.
- Simple example: protocol is a single scene.
- Defined as a set of roles.

LCC Negotiation Scene

```plaintext
negotiate_scene[
  %buyer{
    ...
    }
  %seller{
    ...
    }
]
```
Simplified Example: Roles

- A role is defined as a set of methods.
- Expected to contain an initial method `main()`.
- Roles prefixed with `%`.
- Variables prefixed with `$`.

```
LCC %buyer role
%buyer{
    method(main) = ...
    method(deliberate, $value, $seller) = ...
    method(wait, $value) = ...
}
```
Simplified Example: What’s in a method

**Operations:** Programming flow control constructs

**Actions:** Exchanging messages

**Procedures:** Behaviours of the underlying service
Simplified Example: Operations

Method main()

```plaintext
method(main) =
  waitfor
    ...
  then
  ...
```

- $op_1$ then $op_2$  
  [sequence]
- $op_1$ or $op_2$  
  [choice]
- waitfor $op$  
  [loop until]
- $op_1$ par $op_2$  
  [parallel]
- call $m(...)$  
  [recursion]

Operations can also be actions.
Simplified Example: Message Passing

Message passing actions:

- ... => agent(id, role)  
  [send]
- ... <= agent(id, role)  
  [receive]

Method main()

offer($value) <= agent($seller, %seller)

- Receive a message from agent with role %seller.
- offer is the message performative.
- Parameter $seller gets bound to the ID of the agent taking on this role.
- When message is received by agent (e.g., offer($450)), pattern matching against the protocol leads to binding of $value
Simplified Example: Operations

Method main()

```java
method(main) =
 waitFor
  (offer($value) <= agent($seller, %seller) then
   call(deliberate, $value, $seller))
method(deliberate, $value, $seller) =
  ($newvalue = acceptOffer($value, $seller) then
   accept($value) => agent($seller, %seller))
or
  ...
```

- Decision making: not specified by the protocol.
- Have to call the reasoning procedures of the agent.
- Hence the “lightweight” . . .
Service composition revisited

- Service composition: connecting inputs with outputs.
- How is the meaning of the messages determined?
  - Semantic Web story: share global ontologies.
  - Is this feasible? Scalable?
- LCC alternative:
  - Meaning of messages is determined at a local level, i.e. within an interaction.
  - So only the interacting services need to be able share explicit knowledge about the meaning of messages.
The ‘script’ metaphor

- Centralized control: director (server) determines who gets to say their lines.

- Decentralized 1: each actor their own copy of the script. Synchronization via message-passing.

- Decentralized 2: initial actor has the script, plays their part, and passes on the script as part of a message to the next actor in the scene.
Summary: LCC

- Language for specifying multi agent interactions.
- Defines multiparty dialogues between groups of agents.
- Can implement EI specifications.
- Can use FIPA-ACL and can encode FIPA-ACL BDI semantics.
Reading

- Walton, Chap 6.