

## Multi-Agent System Infrastructures

Yves Lespérance

To allow systems to cope with openness and change, we need a way to allow applications/components to automatically configure themselves to work together. Agent techniques that can help are:

- *matchmaker* and *facilitator* agents, that can match agents needing a service to agents that have the capability to provide it;
- *agent communication languages* that allow agents to communicate without being explicitly programmed for that, and do so at a high level, such as by informing each other of what they know, making requests, promises, etc.; e.g. ICL in OAA, KQML, FIPA ACL;
- *protocols* that allow groups of agents (that may form dynamically) to work together; e.g. contract net — request for proposals, bidding, selection of winning bid, etc.

## Motivation

With the growth in the Internet and electronic commerce, use of computer technology is changing in major ways:

- applications must often operate in open systems, i.e., systems where the set of entities involved can change with time; e.g., hosts go down, mobile computers connect and disconnect, new versions of software become available, etc.
- application have access to a lot of information from multiple sources (e.g. on the Web), but much of it is unstructured (i.e., not from a database with a known schema) and in natural language.

Agents are viewed as a promising technology for dealing with these challenges.

To deal with the second problem, i.e. allowing agents to understand each other even though they use different terminology, mechanisms have been proposed such as:

- *ontology brokers* that make vocabularies (ontologies) for particular application domains available,
- *interpreter agents* that can translate between vocabularies or natural and artificial languages.

## E.g. MAS Platform: OAA

The Open Agent Architecture (OAA) is a platform for developing multi-agent systems (MAS) developed at SRI.

Like many similar tools, OAA has evolved from the *blackboard systems* that have been developed in distributed AI. Problem solving entities (often rule-based) exchange information by putting it on the blackboard.

OAA provides a *facilitator* agent that act as a match-maker. Agents tell the facilitator about their capabilities and send their requests to the facilitator which routes them to an agent that can fulfill them.

OAA agents communicate in the Interagent Communication Language (ICL), a somewhat speech acts-based language with a Prolog-like syntax.

OAA provides libraries for writing agents in Prolog, Java, etc. They support communication with the facilitator.

4

## E.g. MAS Platform: JADE

JADE = Java Agent Development Framework; developed at Telecom Italia Lab.

Compliant with standards of Foundation for Intelligent Physical Agents (FIPA).

FIPA reference model of an agent platform: high-level communication facilities (including with other FIPA compliant platforms using CORBA IIOP), yellow page service with Directory Facilitator agent, agent management.

Communication in FIPA agent communication language (ACL): speech act-based, similar to older KQML, but has formal semantics. Standards for common protocols such as contract net, English and Dutch auctions.

5

## E.g. MAS Platform: JADE (cont.)

JADE is Java-based. Tries to keep communication and concurrency overhead low. On one host, each agent implemented using a single thread within a single Java VM and message passed as objects using Java event (no serialization or conversion to String).

Can have multiple agent containers running on several hosts, with message passing done using Java RMI (requires serialization).

Also communication with other FIPA compliant platforms using CORBA IIOP (requires conversion to String).

Agents don't need to know how communication is implemented.

6

## E.g. MAS Platform: JADE (cont.)

All agents from FIPA reference model are provided, as well as remote monitoring agent with GUI.

JADE provides facilities for implementing agents through class Behavior and its subclasses. Has basic/lightweight concurrency support; non-preemptive round-robin scheduling over behaviors, which *must* relinquish control.

Predefined behaviors include sending and receiving messages, playing roles in protocols such as contract net, etc. Support for use of JESS (Java Expert Systems Shell) for reasoning/deliberation.

Tested in entertainment and travel domains. Most used platform in AgentCities.

7