Software Agents and Multi-Agent Systems

Margaret Lyell, Ph.D.
Explorations Minerva LLC
June 5, 2013
Ann Arbor Computer Society

Outline

- Software Agents in Context
- Standards
- Applications: Some Examples
  - Multi-agent Systems for Simulations
  - Agents and Robots
- JADE: Java Agent Development Environment
  - DEMO

Goals of this Talk

- Introduce the agent software concept.
  - Origins, software process for building agents, application classes.
- Look at standards for agent systems.
  - Standards highlight the features of agency.
- Investigate how an agent would be designed.
  - Simulation example
  - Robotics example
- Look at a currently used open source platform that supports agent execution: the JADE platform.
  - Lower level coding <-> detailed agent design driver.

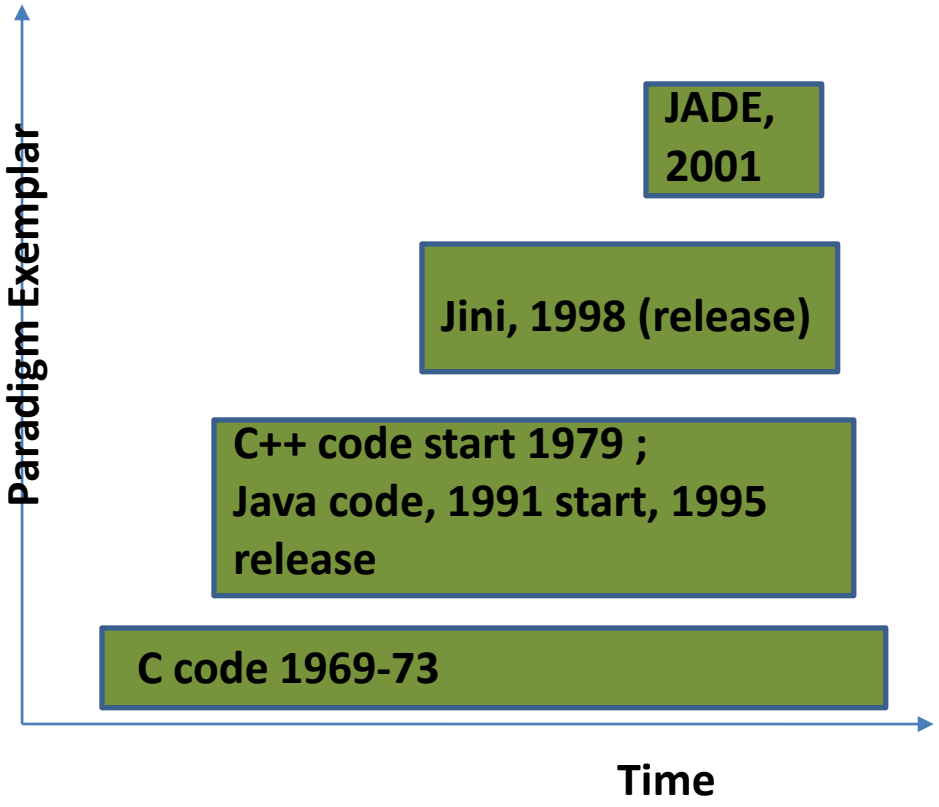
Agents in Ann Arbor

- James Odell
  - Recent Past Chair of Agents Standards Organization, FIPA
  - Author and Editor of numerous proceedings on agent oriented software.
- Dr. H. Van Dyke Parunak
  - Utilizes agent systems in research efforts
  - Agent-based simulation
    - Novel agent concepts
- SOAR Tech
  - Ann Arbor company: http://www.soartech.com/

- Software Agents in Context
- Standards
- Applications: Some Examples
  - Multi-agent System for Simulations
  - Agents and Robots
- JADE: Java Agent Development Environment
  - **DEMO**

Software Agents and Software Paradigms

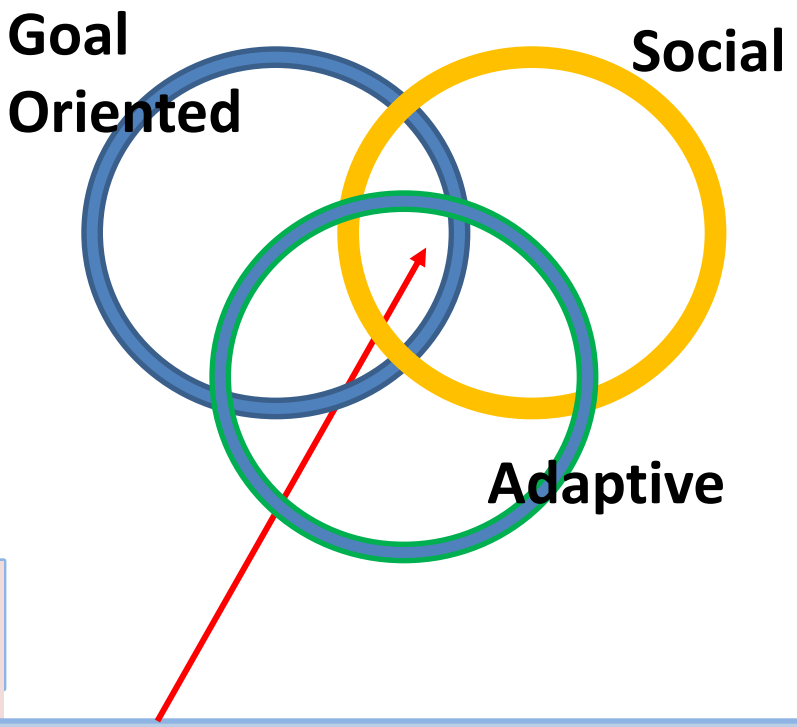
- Procedural
  - C code
- Object oriented
  - Java, C++
- Network –aware
  - Jini
- Multiple, communicating software agents
  - Agent frameworks; e.g. JADE: Java Agent Development Environment (open source)



Dates from Wikipedia except JADE, from Ask.com

What Makes a Software Agent ?

- Software Agent Characteristics
  - Goal/ Task oriented
  - Social / Communicates with other agents
  - Adaptive/ learning



Software Agent went from 'giant brain' to a participant in a Multi-Agent System

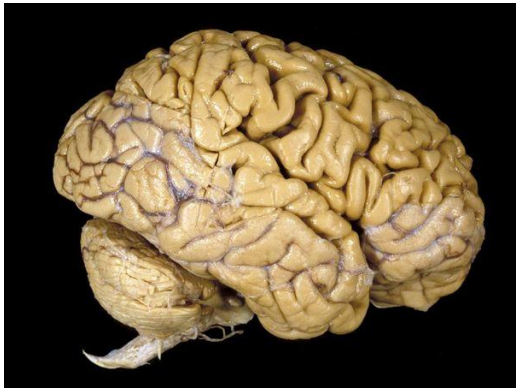
Agent with the 3 characteristics is an Intelligent Software Agent

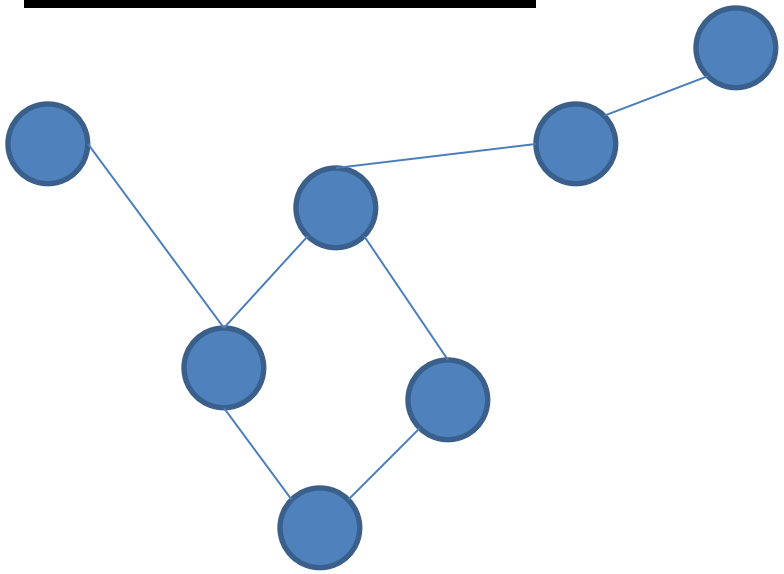
Contributing Background Areas

- There were a number of contributing areas of activity in the 90s that synergistically resulted in agent platform development:
  - Distributed Artificial Intelligence
  - Robotics
  - Distributed computing
  - Software component technologies
  - Computer Supported Cooperative Work
  - …

Agents, Multi-Agent Systems and Frameworks/1

- Agent as a ‘big giant brain’
  - Expert system in an agent
- Agents as a software construct for AI approaches
  - Agents are nodes in a Bayesian Network
  - Agent-to-agent communication is traverse of Bayesian network





Agents, Multi-Agent Systems and Frameworks /2

- **Multi-agent systems characteristics and development**
  - Different "types" of agents
  - This is agent in a role; various standard roles
    - Wrapper, Broker, Finder
    - Coordinator, Middle, Description
  - Message exchange among agents according to a Protocol which supports a 'conversational' exchange:
    - Protocol can be viewed as a FSM with bounded states

- **Multi-agent systems characteristics and development**
  - Development of agent-based system follows an agent –oriented software engineering (AOSE) process .
  - The Gaia AOSE
    - Iterative Analysis and Design phases
  - Gaia Reference: Wooldridge, M., Jennings, N.R., Kinny, D. 'The Gaia Methodology for Agent-Oriented Analysis and Design', Journal of Autonomous Agents and Multi-Agent Systems Vol. 3. No. 3 (2000) 285-312

Agents, Multi-Agent Systems and Frameworks: More on the Gaia Methodology /3

Analysis Phase:

- Produces Artifacts of Roles and Interactions among the Roles [no agents in this phase]
- The Roles have 4 attributes:
  - Responsibilities
  - Permissions
  - Activities
  - Protocols
- The Responsibility of a Role relates to the Role's functionality in the application (post requirements phase)
- Responsibility types
  - Liveness: The tasks the agents must fulfill (as per the overall environment.)
  - Safety: The role must prevent 'unsafe' elements from acting within or affecting the system.

Agents, Multi-Agent Systems and Frameworks: More on the Gaia Methodology /4

Analysis Phase: (continued)

- Permissions state what the Role is allowed to do – or is disallowed from
  - Example: The Role may access XYZ database
  - Example: The Role may access Web service ABC but not Web service DEF unless ABC is non-responsive for some specified number of tries.
- Activities are the tasks that must be performed
  - These are the 'what the agent would do'
  - Part of how the Responsibilities (and Role functions) are carried out
- Protocols provide the Interaction patterns among Roles in carrying out a functionality that is larger in scope than a single Role.

Agents, Multi-Agent Systems and Frameworks: More on the Gaia Methodology /5

Design Phase:

- Artifacts are the Agent Types, the Services Model and the Acquaintance Model
- Roles are mapped into Agent Types
  - One Agent Type may encapsulate one or more Roles
  - The number of instances of an Agent Type
- A Service is the function of an agent; it is derived from responsibilities, liveness and activities that form the role that an agent is carrying
- An Acquaintance Model is the set of interactions among the agents
  - This is the protocol selection to be used among the Agent Types that form the Acquaintance set.

Implementation Phase:

- **NOT part of the Gaia methodology**
- **Broad choice:** Use an agent platform to develop agents most quickly

OR develop all 'lower level agent utility code' [e.g. message queues] yourself

Agents, Multi-Agent Systems and Frameworks/6

- **Application types involving Multi-agent systems**
  - Simulations
  - Frameworks
    - Integration with Legacy Systems
    - Agent-based Middleware
      - E.G. Messaging system with value added
  - Software for Robotics

- **Standards for Multi-agent systems**
  - Foundation of Intelligent Physical Agents (FIPA)
    - Now IEEE/FIPA
    - IEEE Computer Society
  - http://www.fipa.org/
  - Standards related to the application area also relevant.

Agents, Multi-Agent Systems and Frameworks: Example of a Multi-agent System that Integrates with J2EE /7

The Weather Service Agent

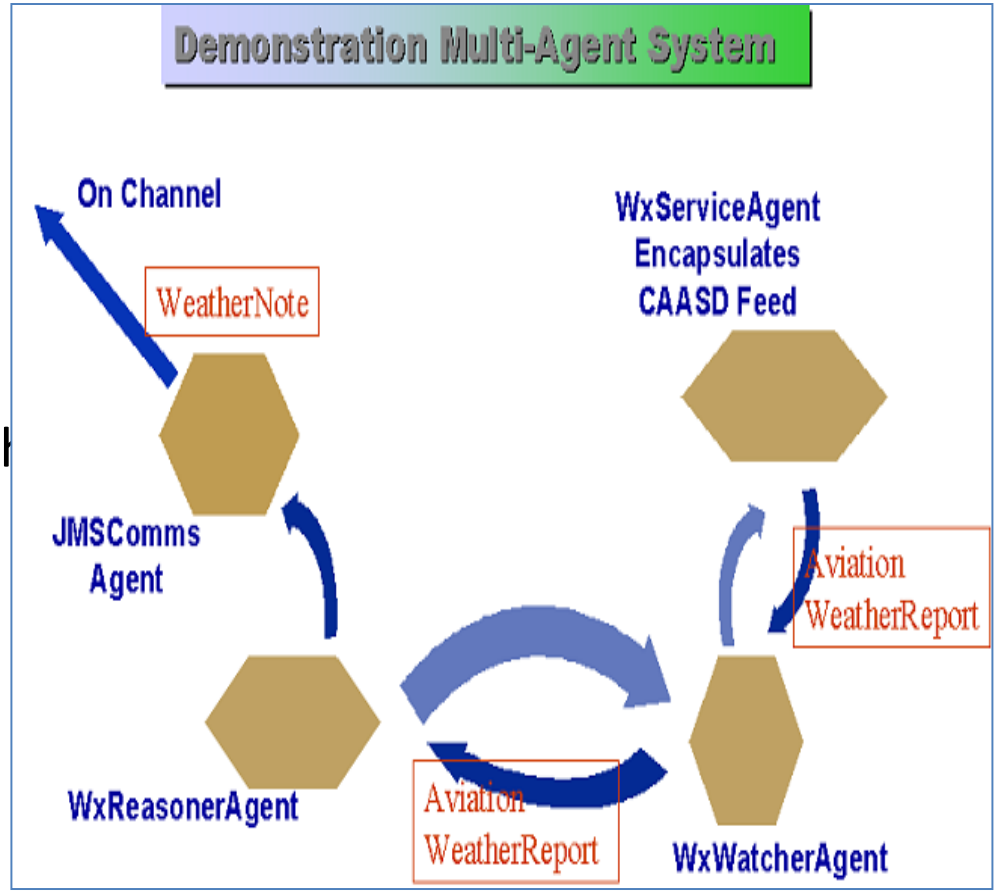
•"wraps" the raw weather feed, providing meteorological aviation reports (METAR) and terminal aerodrome forecasts,

•creates an intermediate product, the Aviation Weather Report, which it then stores internally.

The Weather Watcher Agent

•organizes Aviation Weather Reports along a geographical and time-ordered scheme,

• provides quick retrieval of these reports for agents that ask for them.



Agents, Multi-Agent Systems and Frameworks: Example of a Multi-agent System that Integrates with J2EE /8

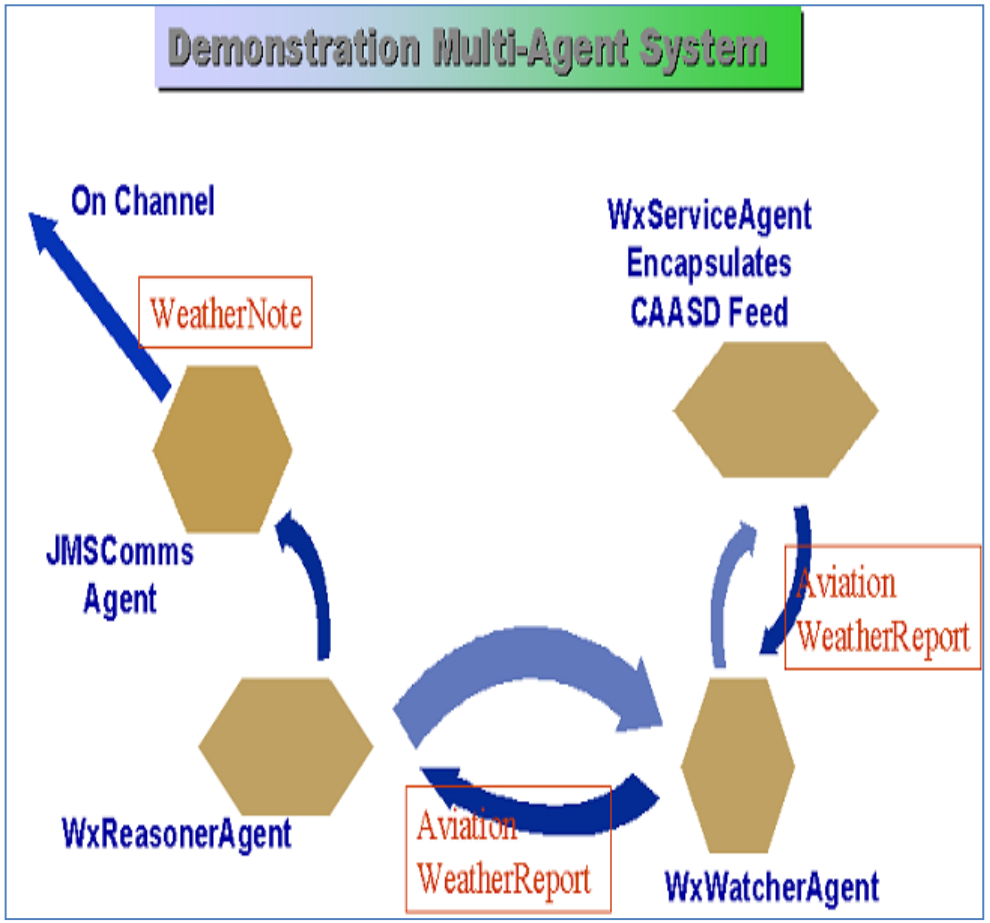
The Weather Reasoner Agent decides if a Weather Note—such as "Fog at Airport XYZ"—is warranted.

It uses the Aviation Weather Reports as input to its decision-making process.

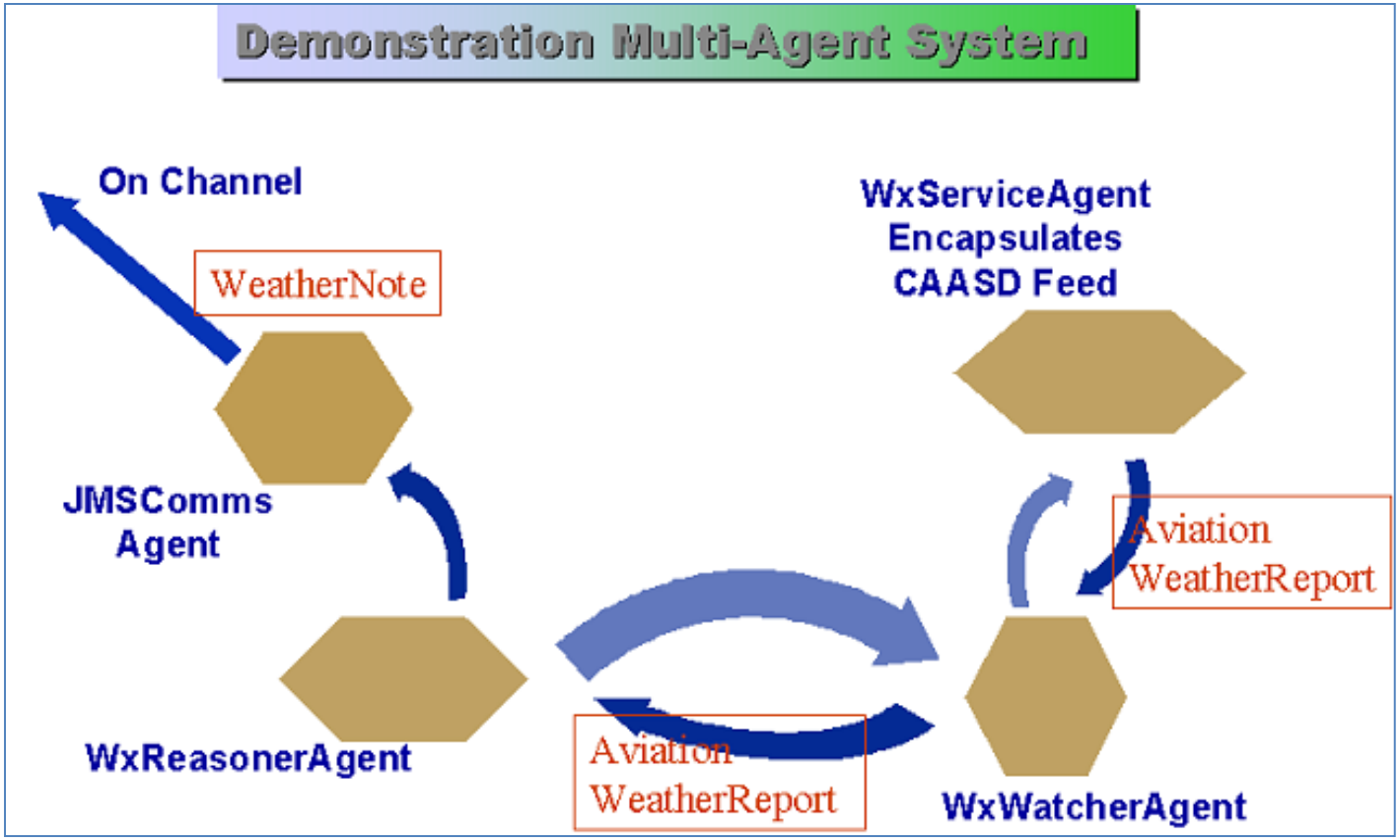
Once the Weather Reasoner Agent has issued a Weather Note, it needs to communicate it to clients –

OFF of the agent platform.

Client is Map client on the J2EE, via JMS COMMS Agent.



Agents, Multi-Agent Systems and Frameworks: Example of a Multi-agent System that Integrates with J2EE /9



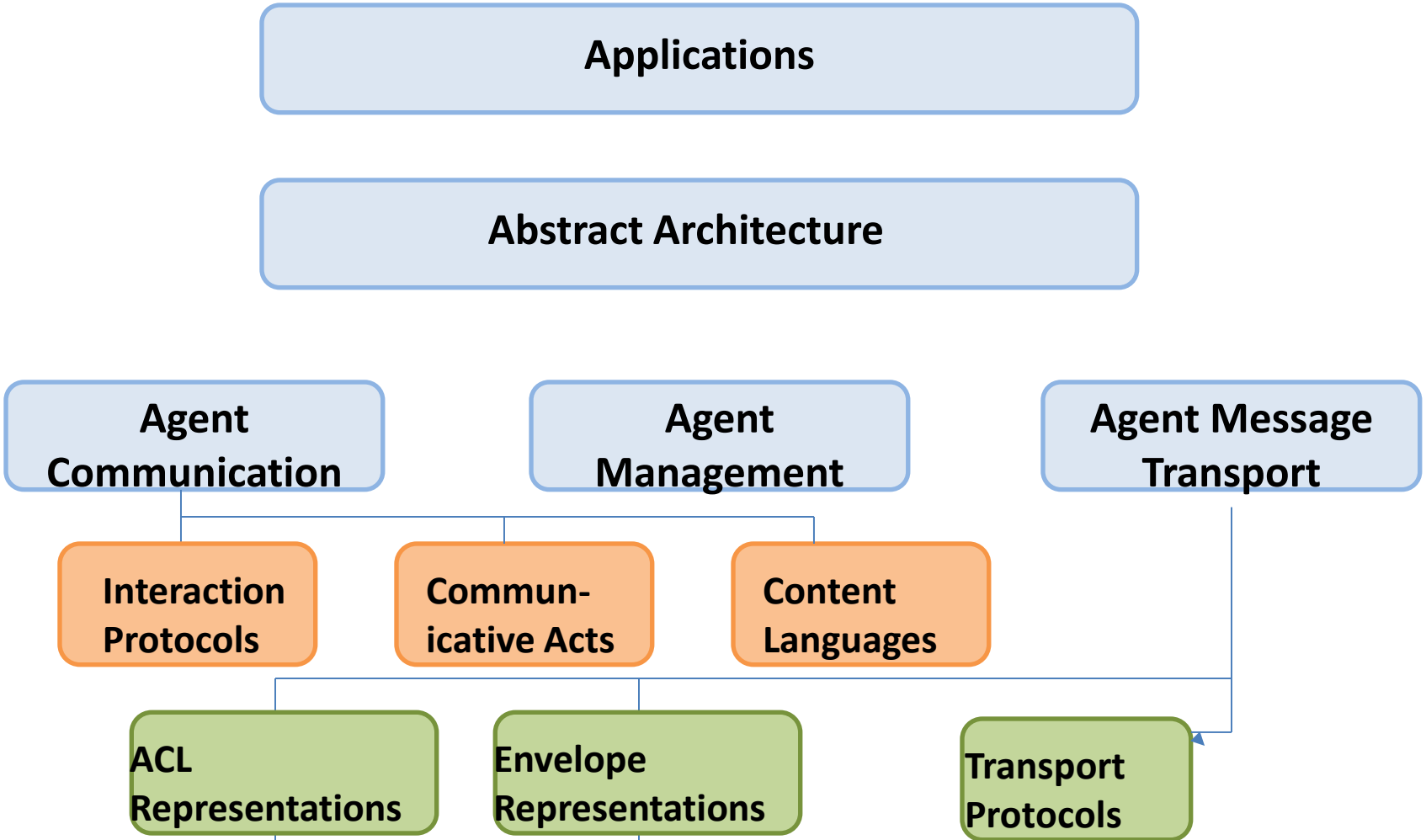
Open Source FIPA-OS Agent Framework Used.

Outline

- Software Agents in Context
- Standards
- Applications : Some Examples
- JADE: Java Agent Development Environment
  - DEMO

Standards for Intelligent Physical Agents

- IEEE/ FIPA: http://www.fipa.org/
- Specifications organized into five groups; ~ 2002



Standards for Intelligent Physical Agents

General Comments on FIPA standards Re usage:

- Situates an Agent with respect to needed services
  - Elucidates requirements that would be useful on an agent platform
  - Example: directory service
  - FIPA standards allow third party services, the agents do not have to be residents of a platform.
- Interoperability
  - Traction in areas of Message Structure, Protocol Interactions
    - **More on Message and Protocols**
  - Message Content: Language in FIPA specification is SL; the profile “SL0” most used
  - Experimental profiles for RDF, KIF (Knowledge Interchange format)

Abstract Architecture: Agent Relationship (UML Diagram)

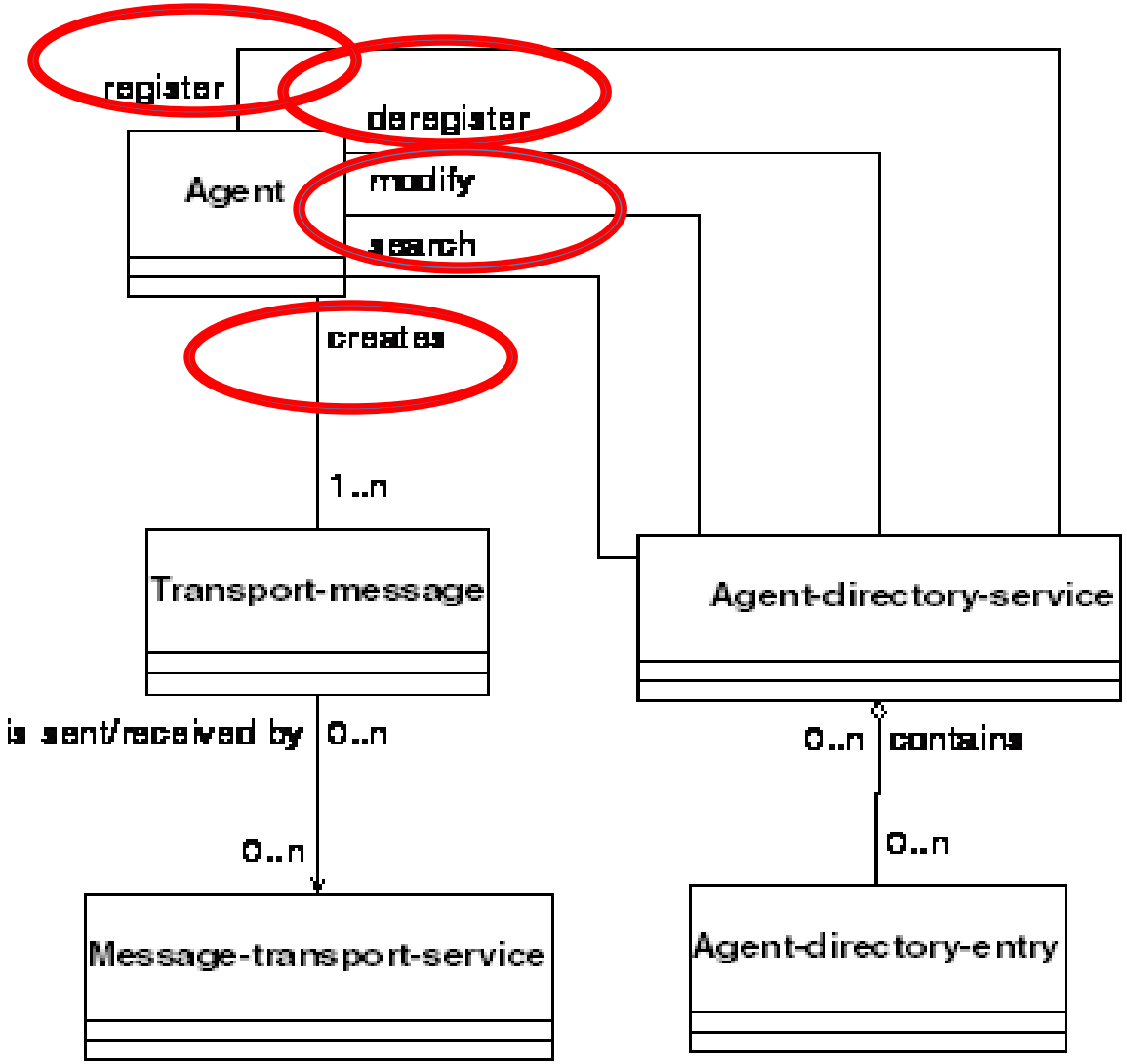
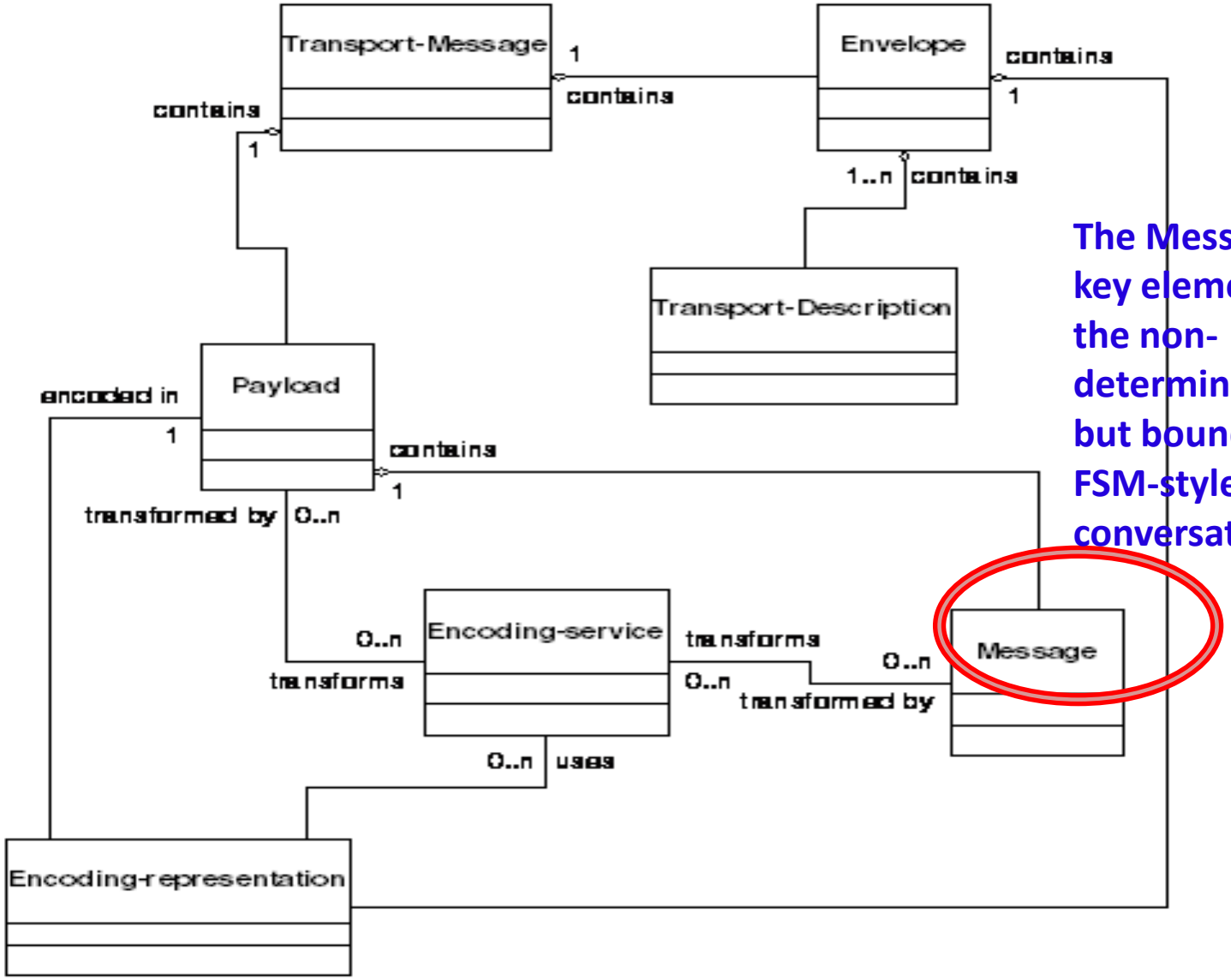


Diagram from FIPA Specification at Web-site http://www.fipa.org/

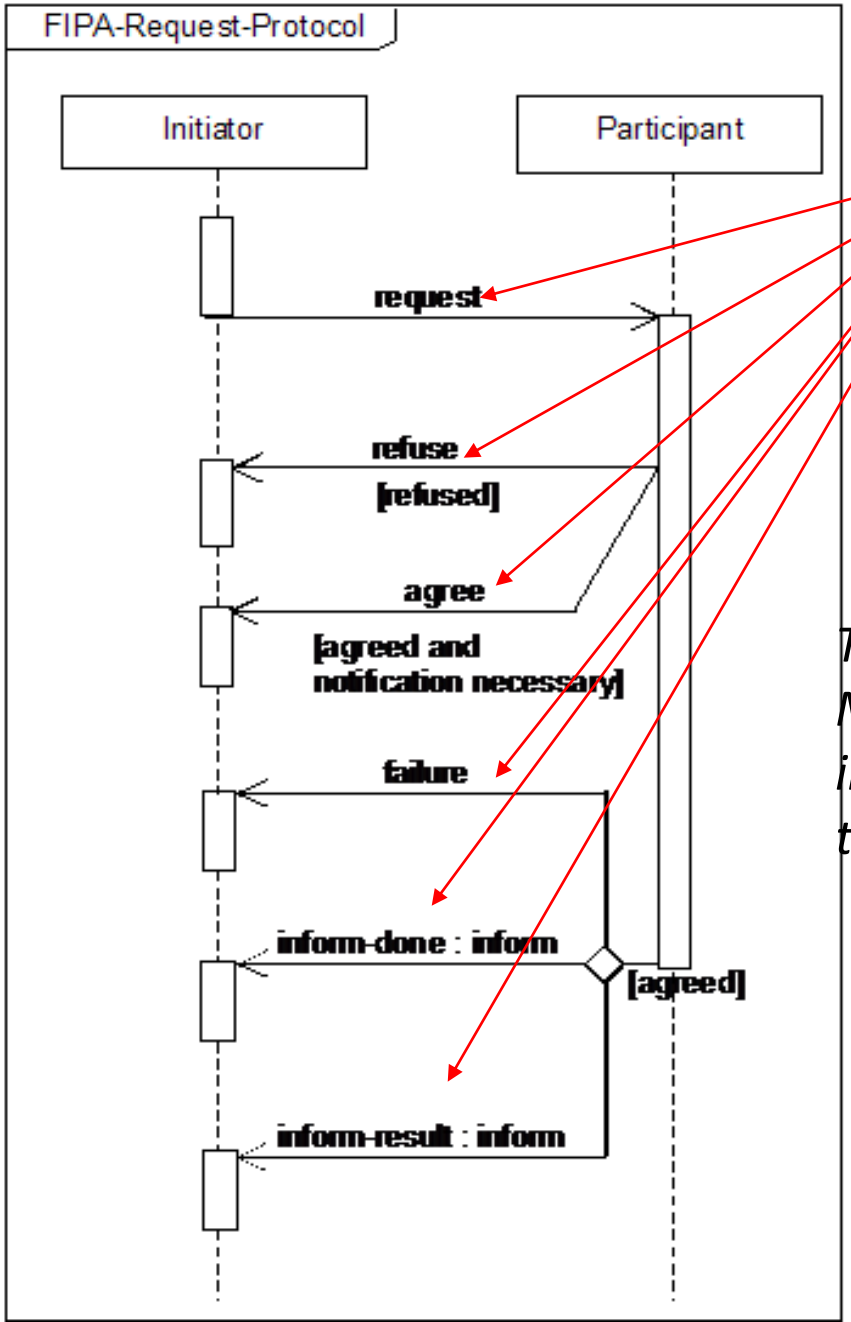
Abstract Architecture UML Diagram: Message and Transport Message



The Message is a key element of the non-deterministics but bounded FSM-style conversations.

Diagram from FIPA Specification at Web-site http://www.fipa.org/

Interaction Protocol: Request



Performatives

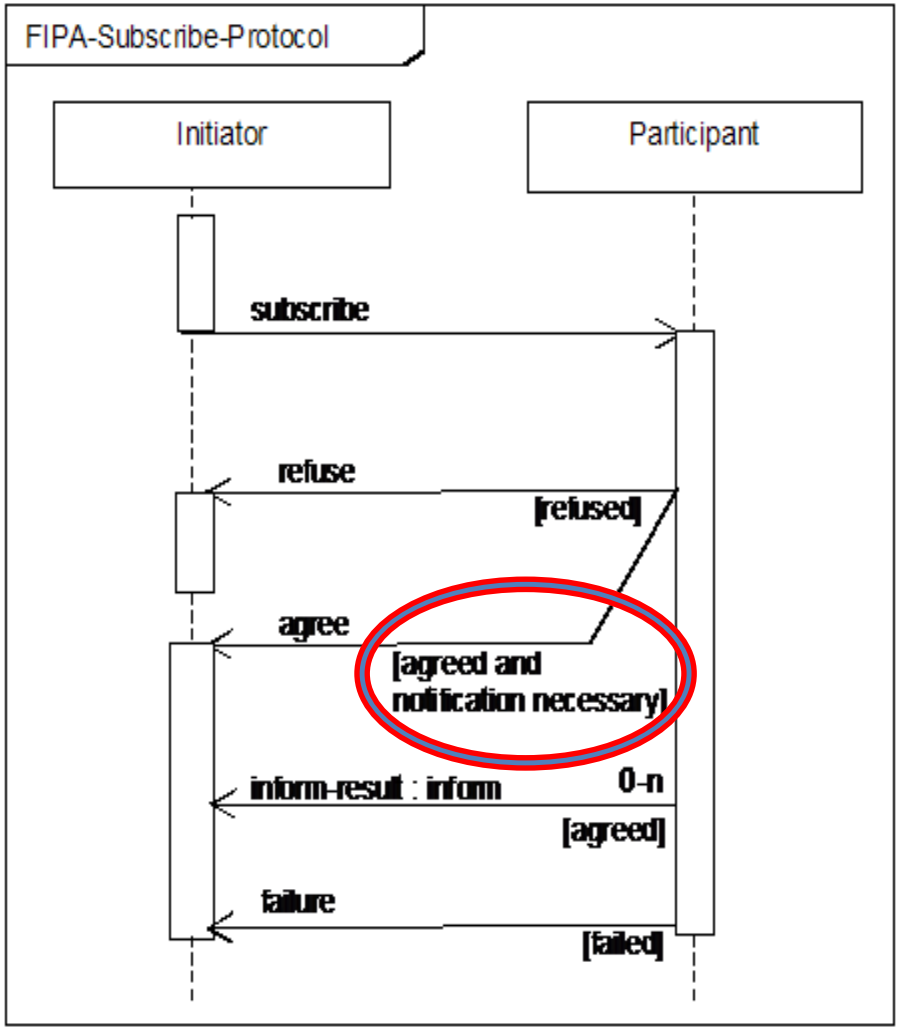
The choice of an initial response sends the interaction down a certain path: non-deterministic but finite options.
E.G. {Request, Agree, Failure}
{Request, Refuse}
Earlier version had a 'do not understand'

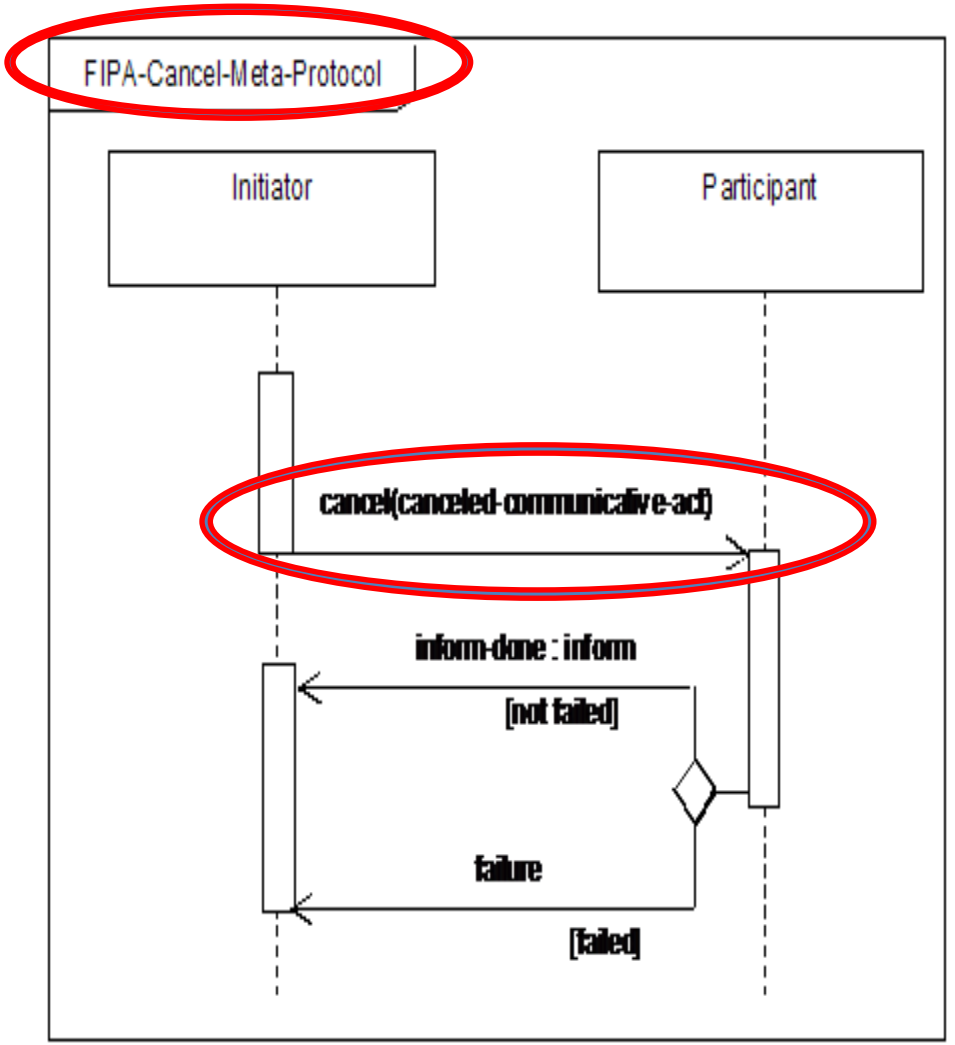
The 'failure' also serves as an ACK; Messages are asynchronous and the initiator has no infrastructure imposed time-out.

Concept is that the agent is performing a speed act. *The performatives indicate actions (due to illocutionary force).*

Diagram from FIPA Specification at Web-site http://www.fipa.org/

Interaction Protocols: Subscribe and Cancel





Diagrams from FIPA Specification at Web-site
http://www.fipa.org/

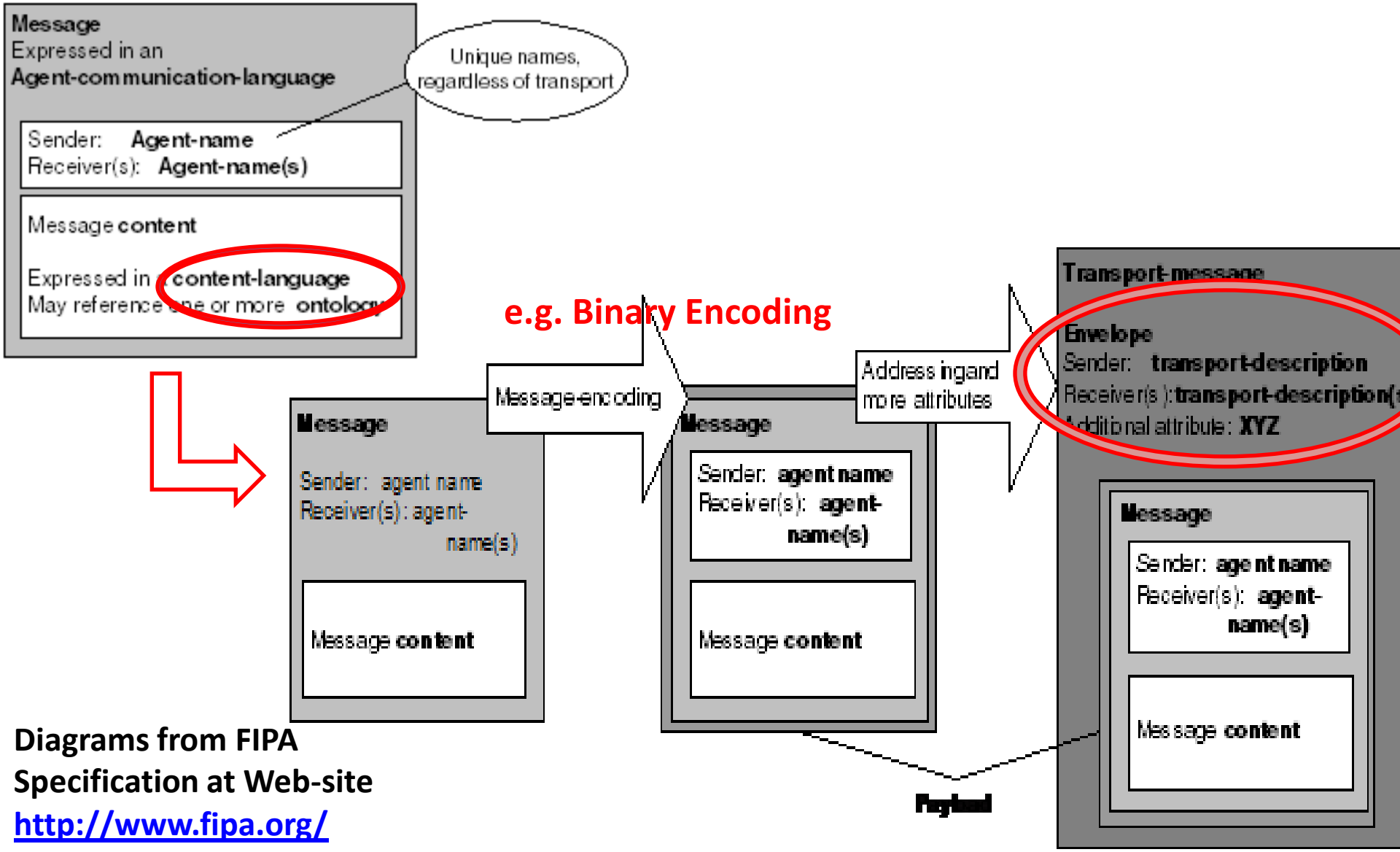
Messages: FIPA-compliant Agents vs. J2EE Messages

Conceptual Mapping of Message Related Entities between FIPA and JMS

FIPA	JMS (J2EE)
FIPA Message	
FIPA Envelope	Aggregate JMS Message Headers
FIPA Envelope {name,value} pairs	JMS (optional) Message Properties
FIPA Transport-Message	JMS Message
FIPA Transport-Message Payload	JMS Message Payload

Agents bring flexibility to system design and offer loose coupling alternatives for system interoperation.

Abstract Architecture: Message to Transport Message



Diagrams from FIPA Specification at Web-site
http://www.fipa.org/

Agent Communication Language

FIPA SL0

- Profile 0 of FIPA SL is the minimal subset of the FIPA SL content language.
- It allows:
  - representation of actions,
  - determination of the result a term representing a computation,
  - completion of an action,
  - simple binary propositions

Outline

- Software Agents in Context
- Standards
- Applications: **Some Examples**
  - Multi-agent System for Simulations: Pedestrian agents
  - Agents and Robots
- JADE: Java Agent Development Environment
  - DEMO

Developing Simulations with the Pedestrian Agent

There have been simulations of crowds and traffic analysis:

- Traffic Engineering
- Urban Design
  - Concourse, stadiums; City areas; Buildings
- Crowds and pedestrian motion (Sill, 2000)
- Pedestrian Evacuation
  - Rooms; Public Spaces; Airplanes; Ships

Concept of using a Pedestrian Agent in simulations

Possible features in building the 'full' Pedestrian agent concept:

- Locomotion should be realistic
- Pedestrian agent should have emotional qualities
- Pedestrian agent should have rational qualities
- Pedestrian agent should be goal oriented
- Pedestrian agent should have a belief set

Designing the Pedestrian Agent

Design Elements

- Walking Rules (Blue and Adler)
  - From the Transportation Literature
  - Blue and Adler provided a set of rules for how pedestrians walk, based on experiments and observations
  - These rules can be modeled using cellular automata
  - Blue and Adler did Not Incorporate Wayfinding
    - Goal Driven Pedestrian Agents should also utilize way finding
- Cognitive Structure of Emotions: OCC Model (
- Personality Model: Five Factor Personality Model (Digman)
- Beliefs
- Cognitive Capabilities
  - Goals
  - Action Sets

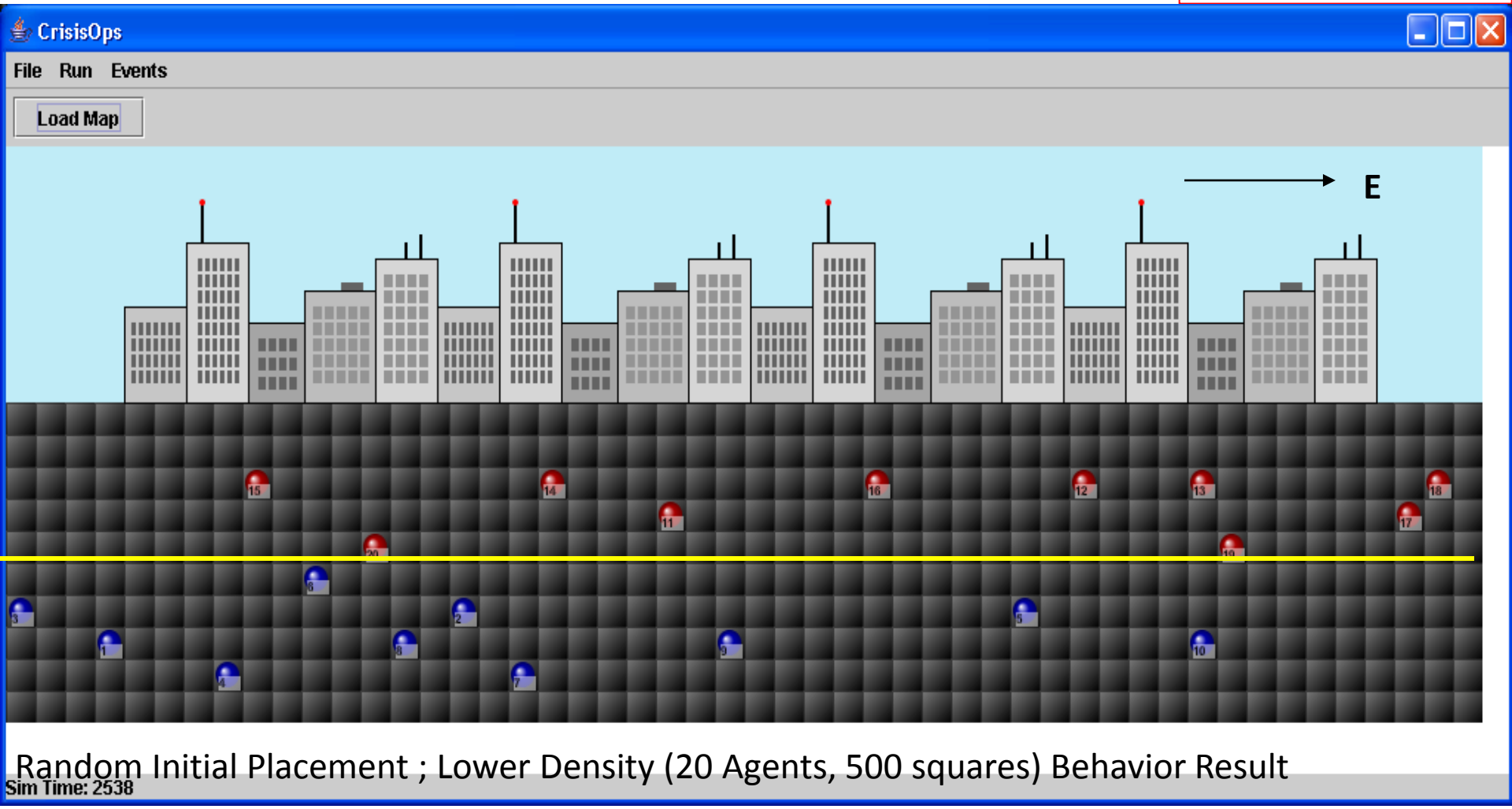
Building the Agent-based Simulation

- **Relevant prior work included:**
  - Walking rules (not in context of agents)
  - Overall crowd motions (Sill, 2000)
    - 'Pre-agent'

- **Process of Building the Simulation:**
  - Use the walking rules and basic goal concepts cast into a 'not-really cognitive' Pedestrian agent: convert CA to agent usage.
  - Add the cognitive and emotional elements to the Pedestrian agent – in as simple as manner as possible to start.
  - Improve the cognitive and emotional modeling, add belief structure.
  - Improve the simulation framework: Make it user friendly.

Emergent Behavior: Spontaneous Lane Formation

Recovers Blue and Adler results

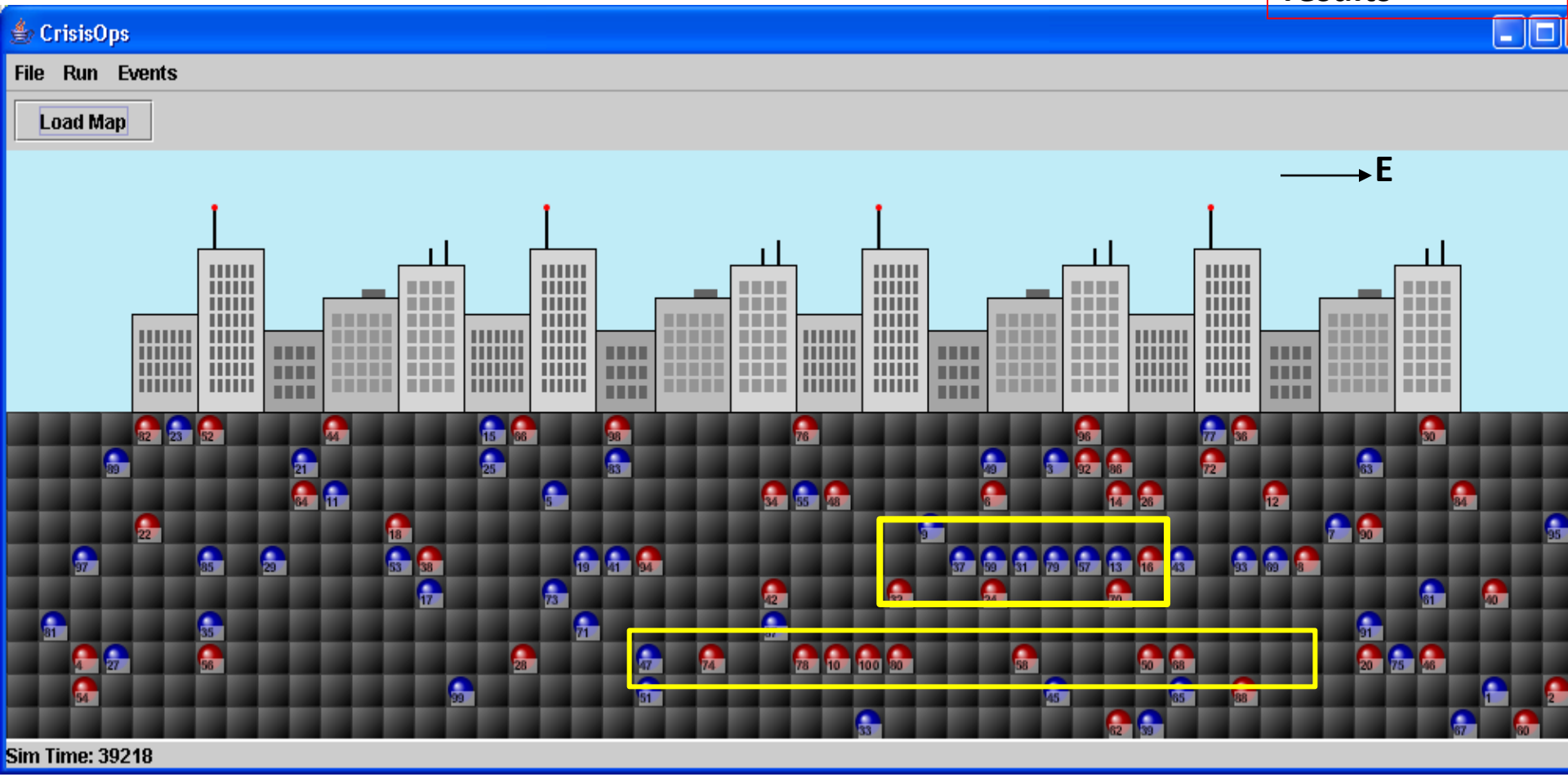


Random Initial Placement ; Lower Density (20 Agents, 500 squares) Behavior Result

- WEST-ward moving pedestrian, stroll, no cognition
- EAST-ward moving pedestrian, stroll, no cognition

Emergent Behavior: "Partially Organized Lanes" at Higher Densities

Recovers Blue and Adler results

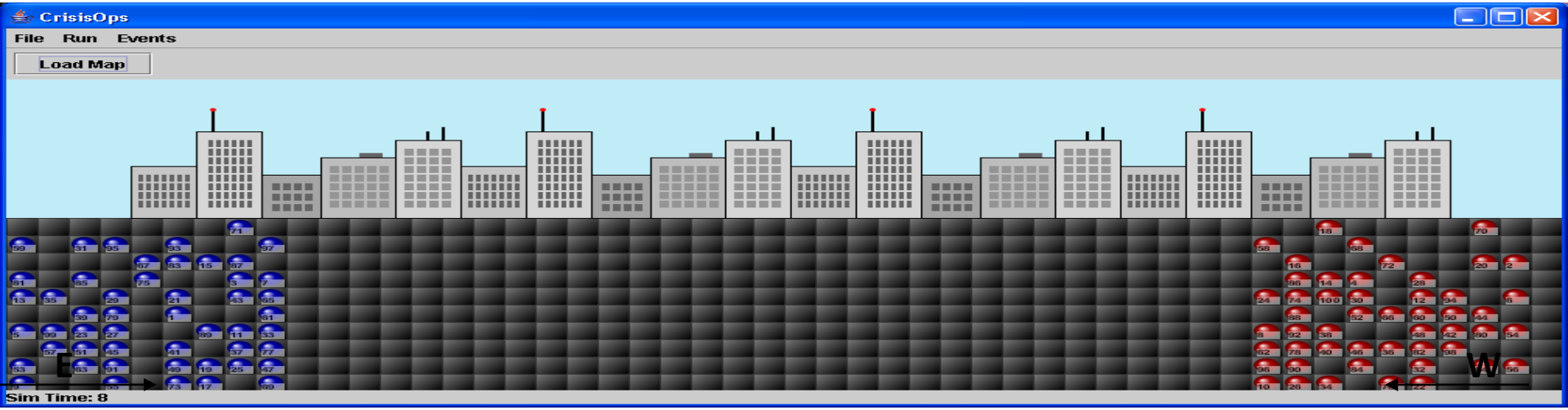


Random Initial Placement ; Density = 0.20 (100 Agents, 500 squares) Behavior Result

- WEST-ward moving pedestrian, stroll, no cognition
- EAST-ward moving pedestrian, stroll, no cognition

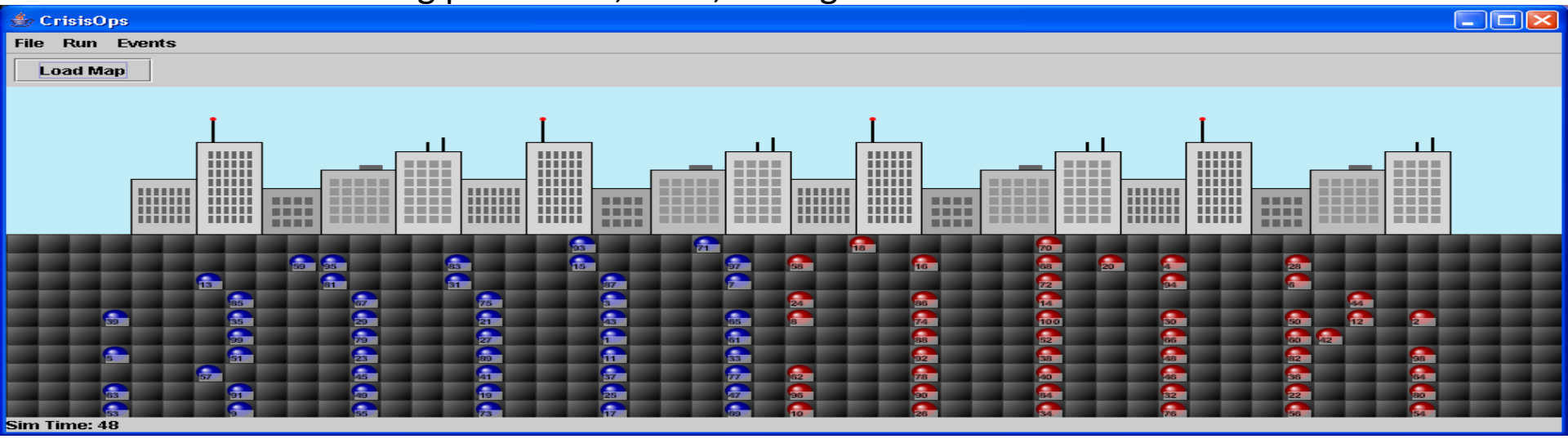
Simulation Scenario: Crowds Moving Through Each Other

Crowds Moving Through Each Other: Initial Pedestrian Distribution

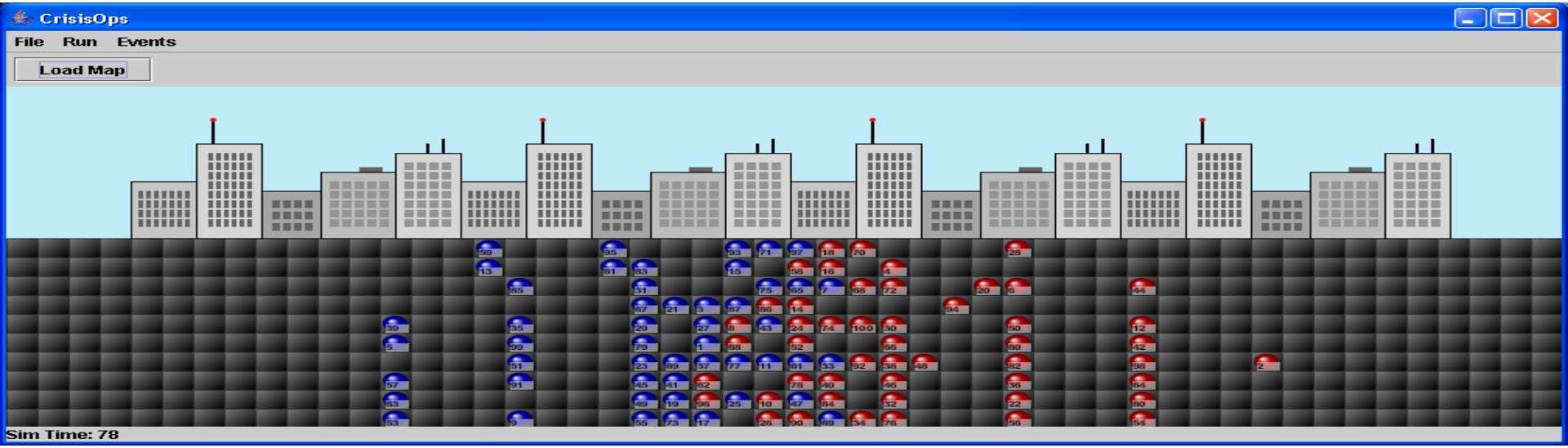


Density = 0.20 100 Agents, 500 squares on Grid

- WEST-ward moving pedestrian, stroll, no cognition
- EAST-ward moving pedestrian, stroll, no cognition



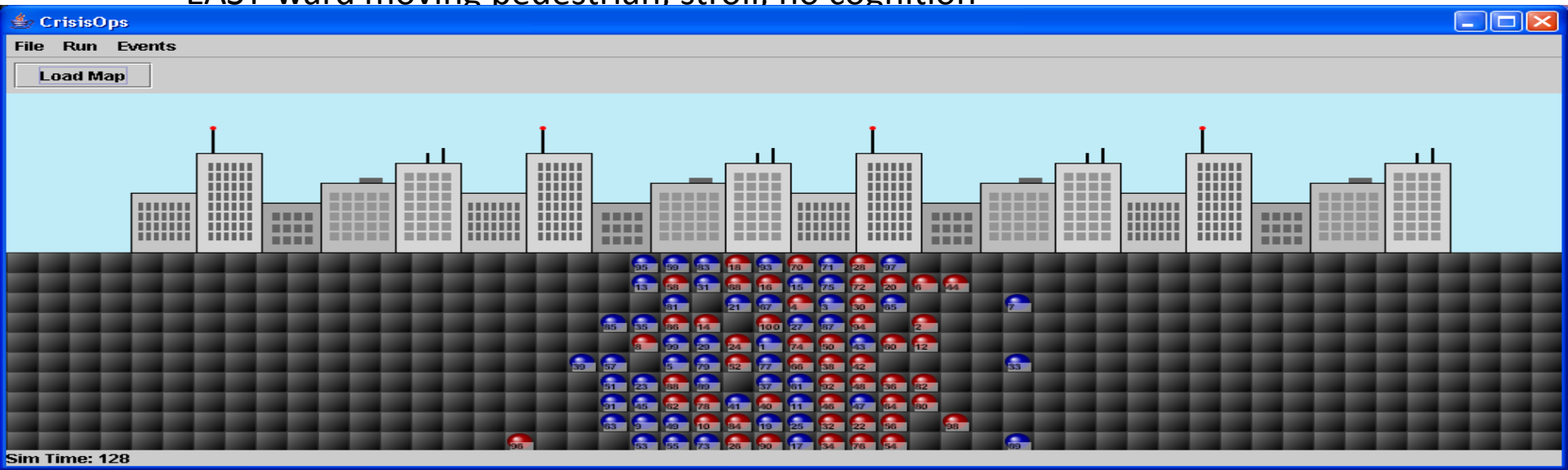
Crowds Moving Through Each Other: Opposite Motion Crowds Meet



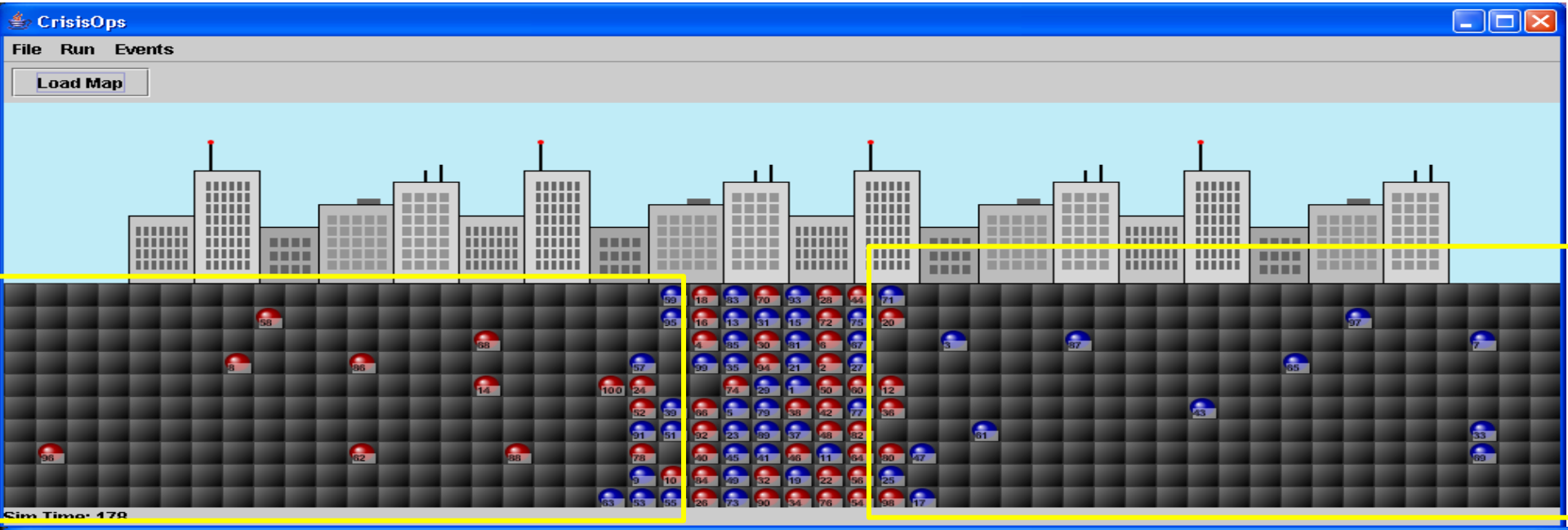
Density = 0.20 100 Agents, 500 squares on Grid

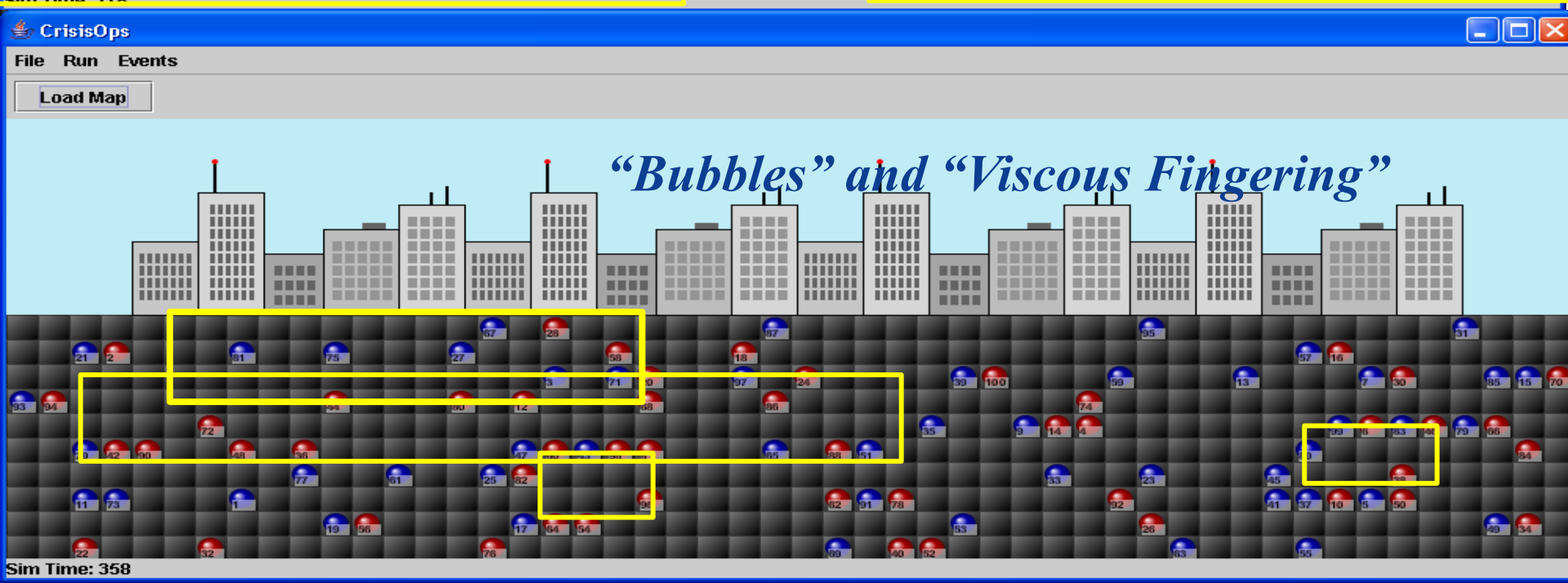
WEST-ward moving pedestrian, stroll, no cognition

EAST-ward moving pedestrian, stroll, no cognition



Opposite Motion Crowds Moving Through Each Other





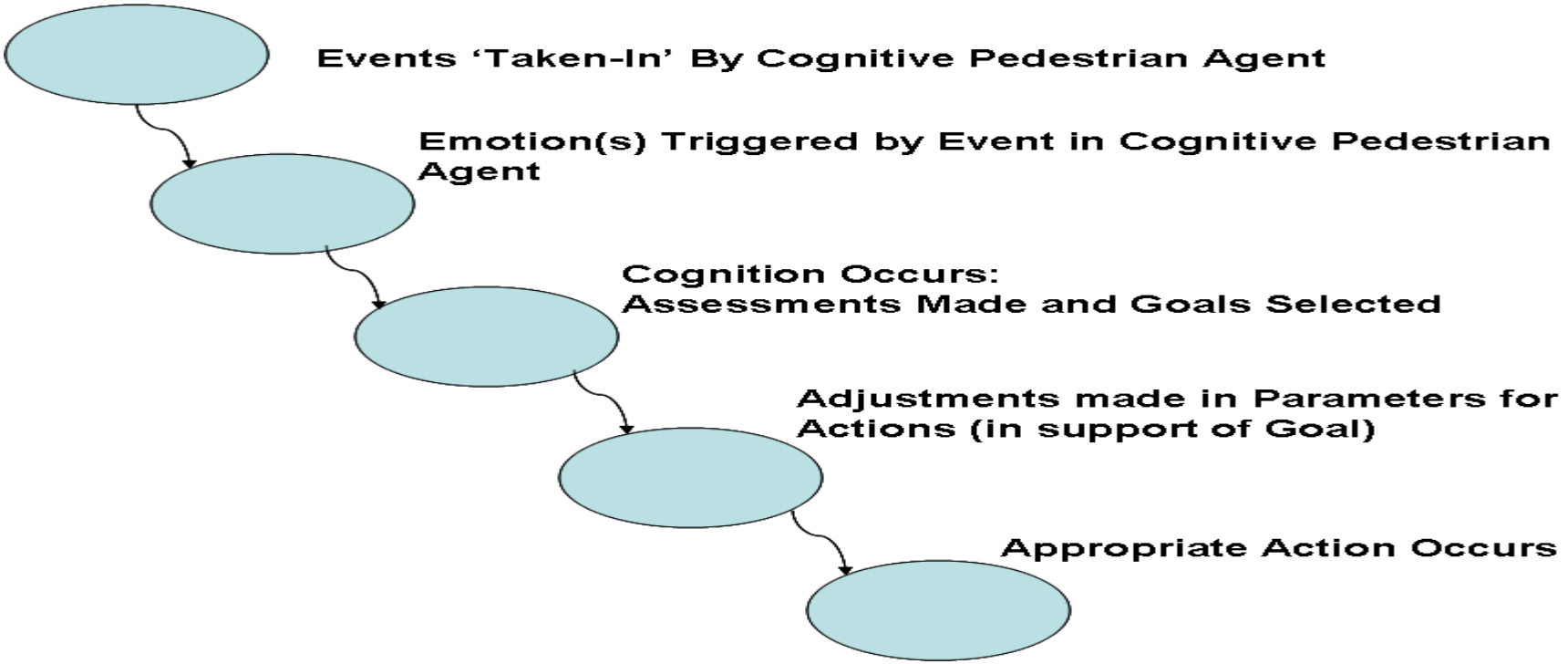
The OCC Cognitive/Emotion Model and the Five-Factor Personality Model

OCC Cognitive/Emotions Model:
•Ortony, Clore, Collins, 1988, "Cognitive Structure of Emotions"
•Emotions arise from (positive or negative) reactions to situations consisting of (1) Events, (2) Objects and (3) 'agents'.
•*Emotions play a role in cognition and affect actions that are taken*

OCC Framework Details
•List of 22 emotions (other variations list fewer or more)
•Rule based system for generating the emotional types ➔
Computational representation

Digman Five-Factor Personality Model
•1990 Personality structure: Emergence of the five-factor model. *Annual Review of Psychology, 41,* 417-440.
•*Effect of emotions are filtered through the personality type*

Cognitive Pedestrian Agent: Processing Cycle



Cognitive Agent Personality Types:

Caricatures

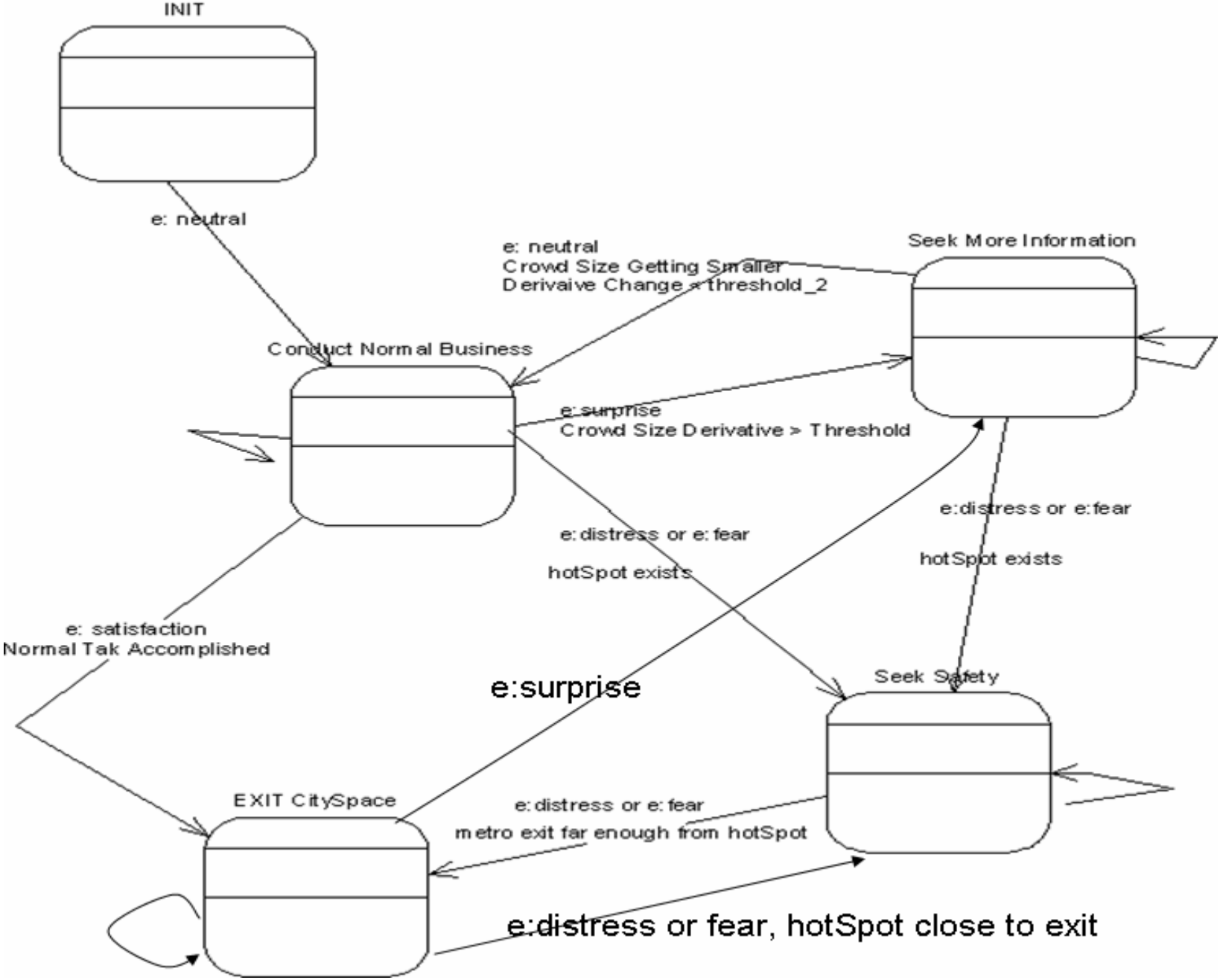
- Fearful Pedestrian Agent
- Curious Pedestrian Agent
- Social, Agreeable Pedestrian Agent

Locomotion Rules + Cognitive Agent with Personality

Caricature Cognitive Pedestrian Agent Types in Fire Crisis – All Cases

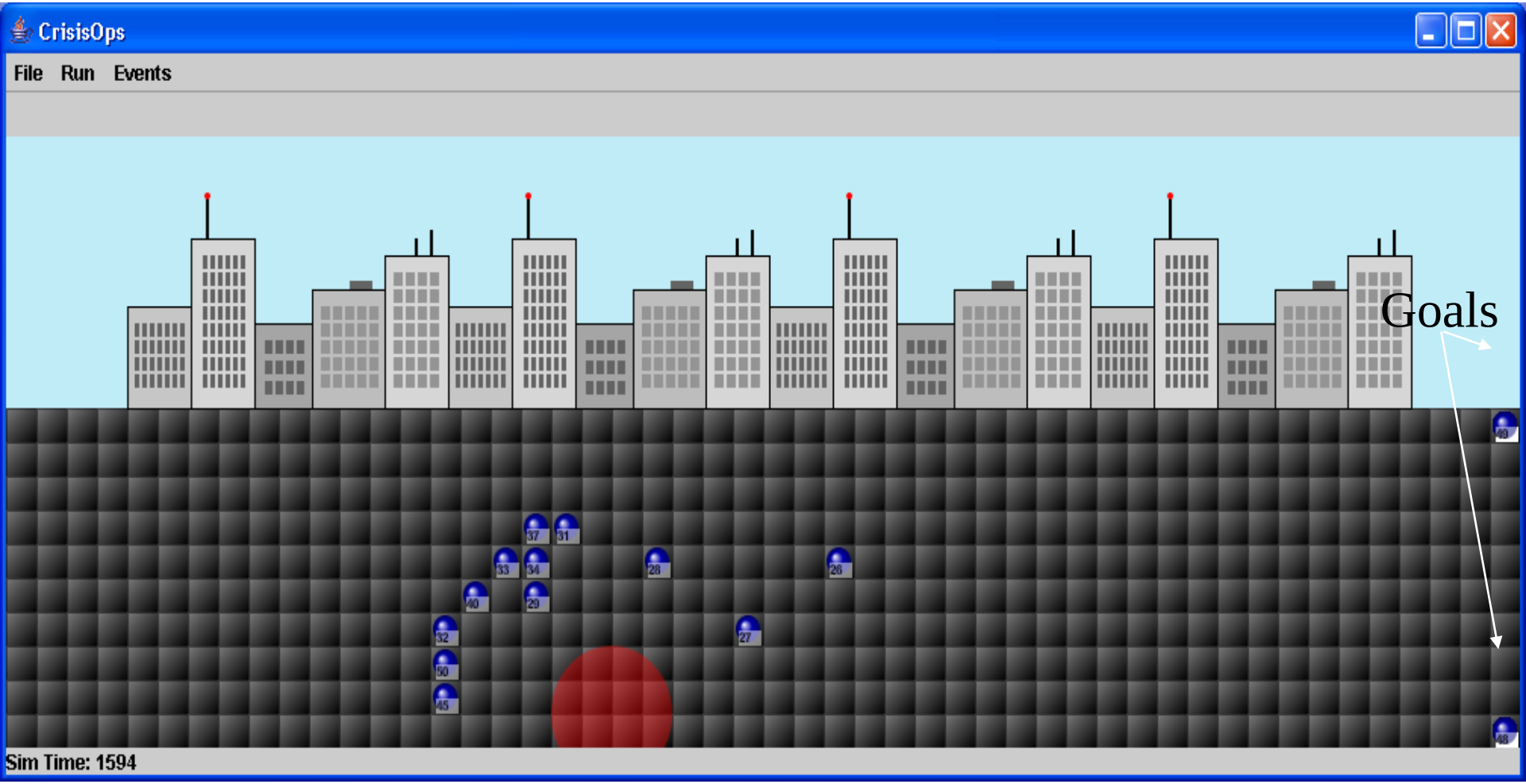
- Fearful Agent
  - If learns of fire, will seek safety
  - Never pro-actively helpful with ‘fire exists’ messages
  - Will infrequently respond to direct questions from other agents
- Social/ Agreeable Agents
  - Most complex emotional range
  - Proactively helpful – send ‘fire exists’ messages to nearby agents
- Curious Agents
  - Aware of smaller-sized crowds
- All agent types obey police officer directive to leave area
  - Officer in all cases, moves towards fire, orbits fire, redirects adjacent pedestrians
- City Area Description
  - City Grid 10 cells high, 50 cells wide
  - Agents enter left at metro
  - Business Goals at upper and lower city edge (right)
  - Fire Radius 2, Fire Appears at SimTime 200

GOAL STATES: FEARFUL PEDESTRIAN AGENT



Curious (Cognitive) Pedestrian Agent Movement:

Anticipated Agent Component Behavior: Curious Agents Watch Fire

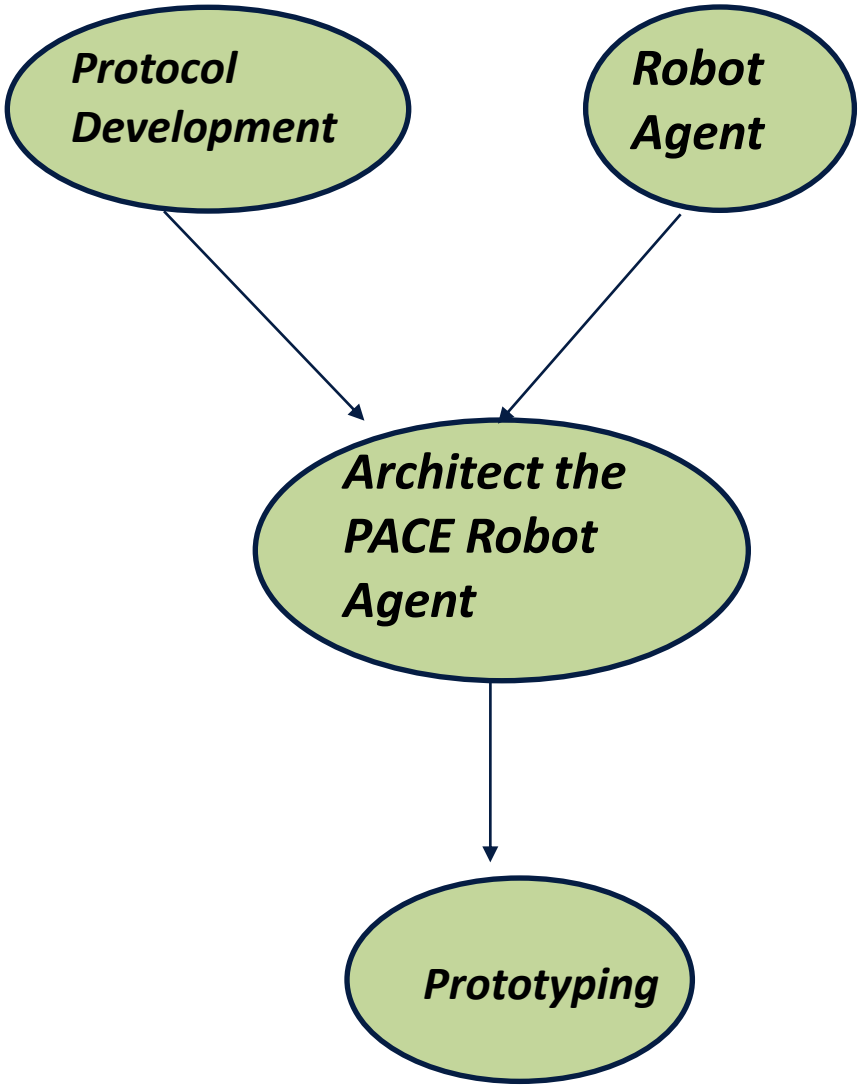


Outline

- Software Agents in Context
- Standards
- Applications: **Some Examples**
  - Multi-agent System for Simulations: Pedestrian agents
  - Agents and Robots
- JADE: Java Agent Development Environment
  - DEMO

Building a Robot Agent used in Autonomous Teamwork

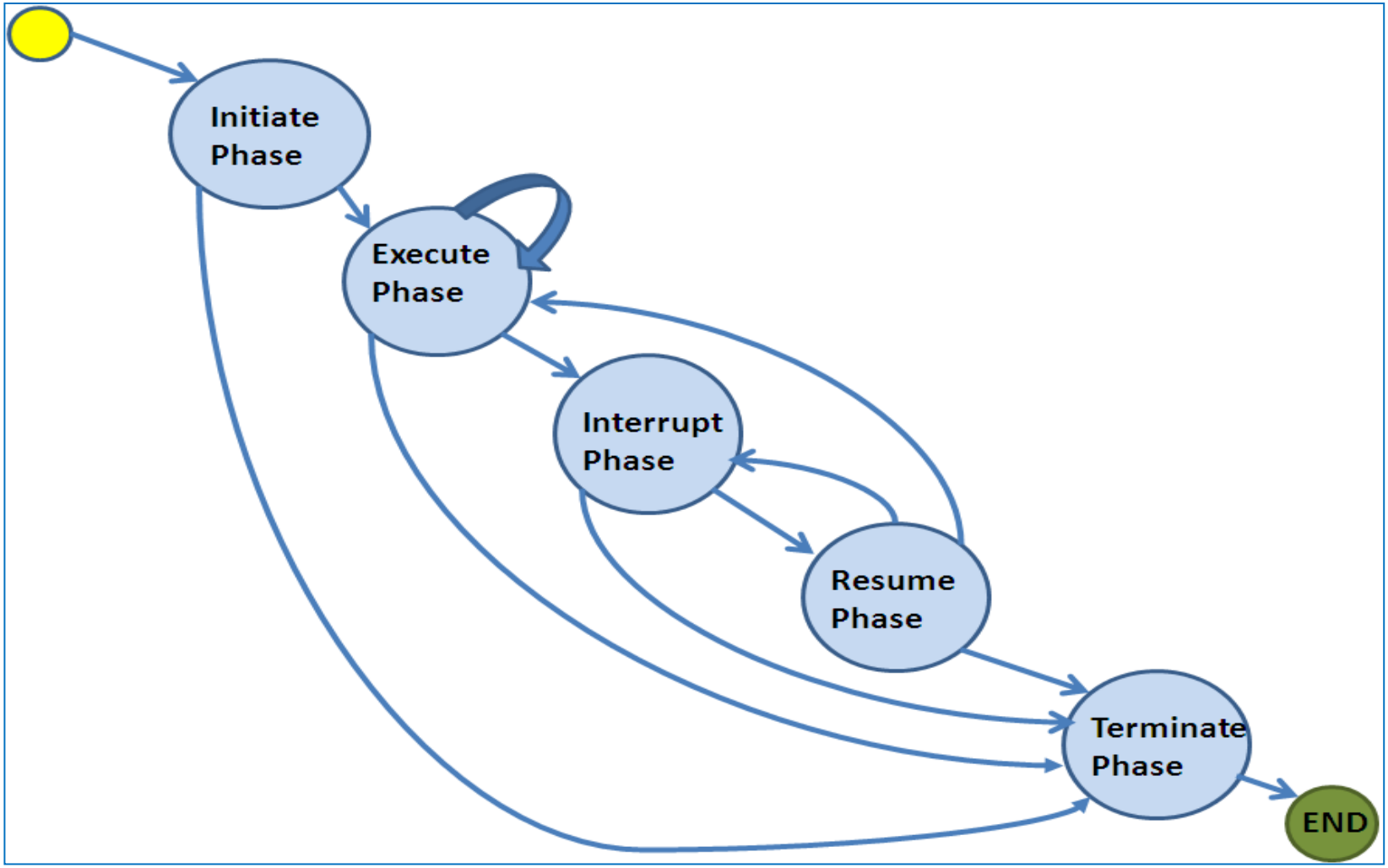
- Protocol Development
  - Develop the Autonomous Team Work Effort Protocol
    - Develop supporting 'mini-protocols'.
- Extend the Robot Agent
  - Architect the Protocol Aware, Cognitively Enhanced (PACE) Robot Agent
- Prototype the Robot Team:
  - Robots in a Scientific Sampling Mission Using the Team Work Effort Protocol



Background for the PACE Robot Agent

- Goal is to insert protocol-awareness into Robot agent capabilities.
  - NASA Lunar Exploration Missions
  - Autonomous Team work Effort needed
  - Robot Agent has a Role (Team Lead, Worker)
- Use Case for PACE Robot Agent:
  - Autonomous team effort to retrieve pre-surveyed area rock samples,
  - The “Lunar Scientific Sampling Mission”.

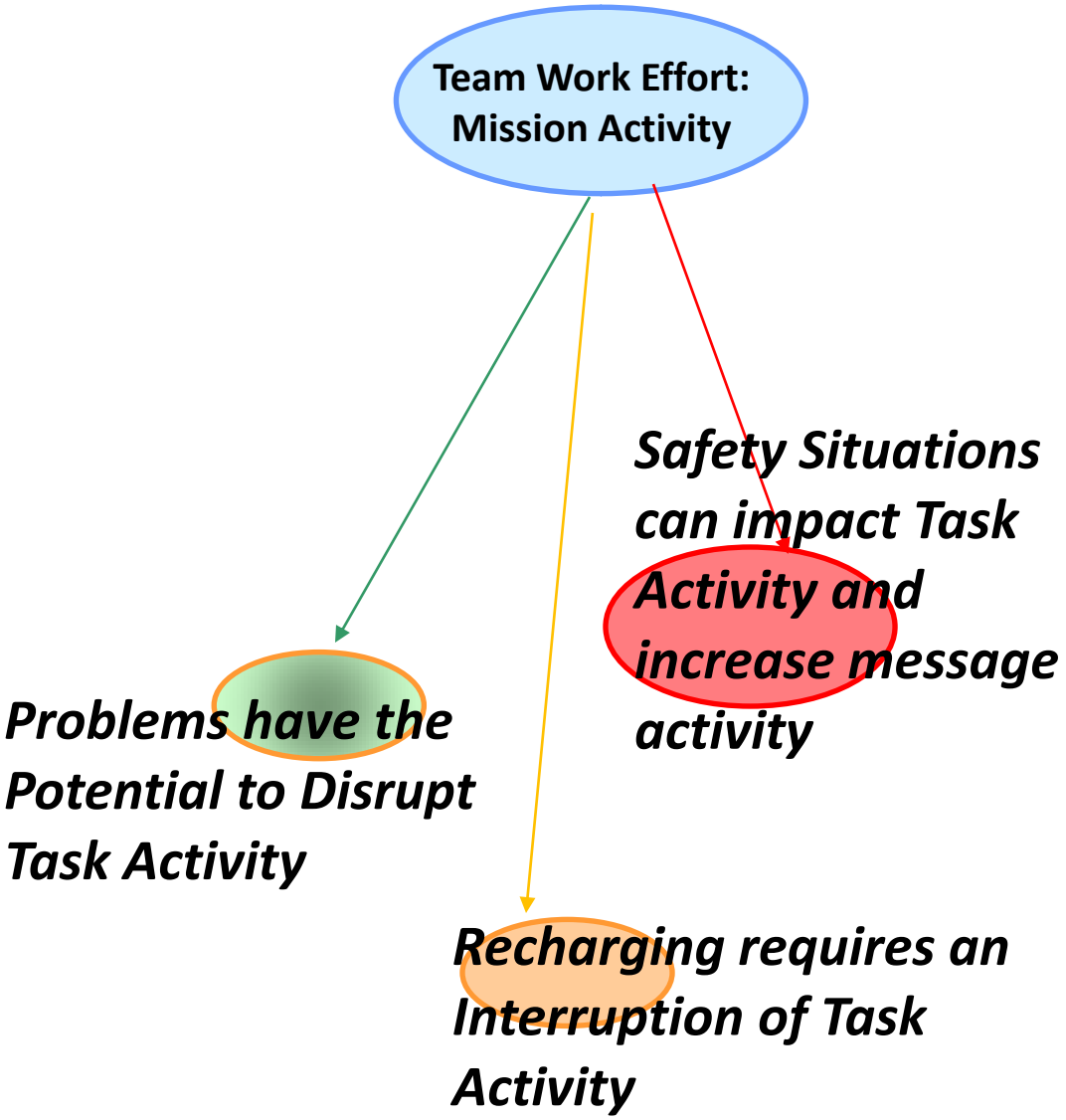
Protocol Development: Finite state machine view of the Team Work Effort protocol



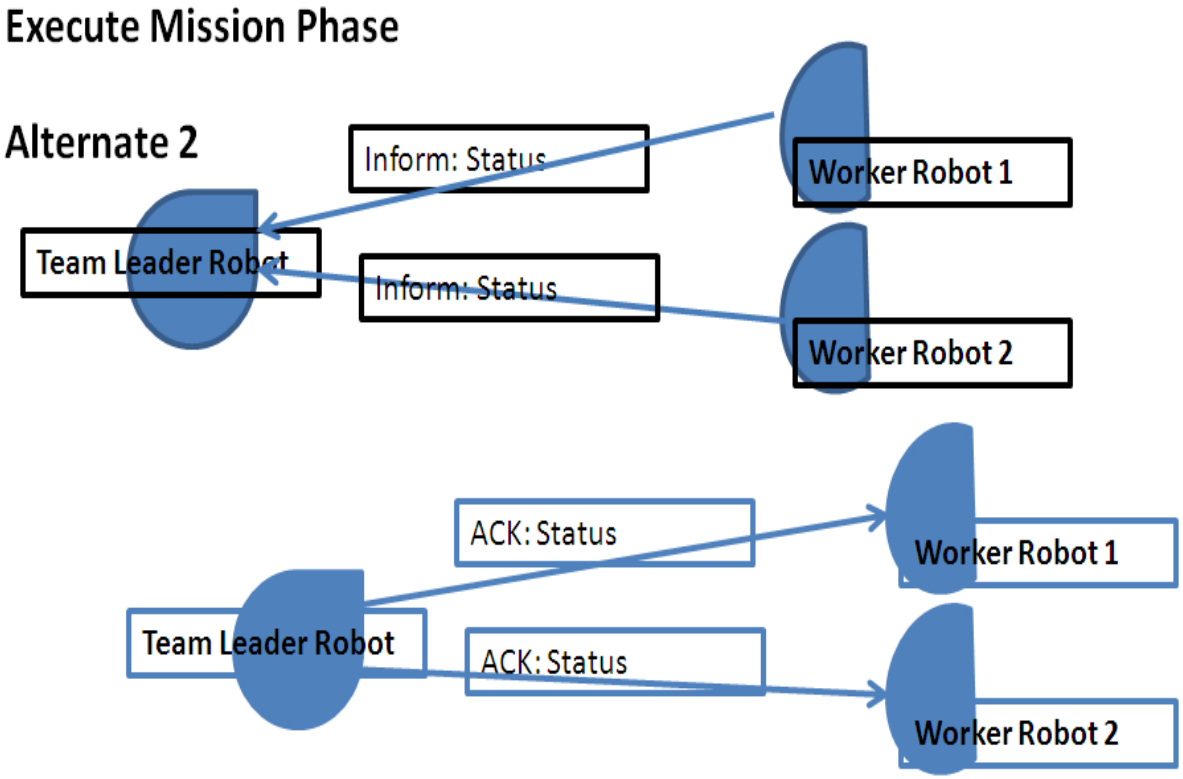
Comments on Details of TWE Protocol Phases

The TWE Protocol was developed iteratively: factored out the Problem protocol

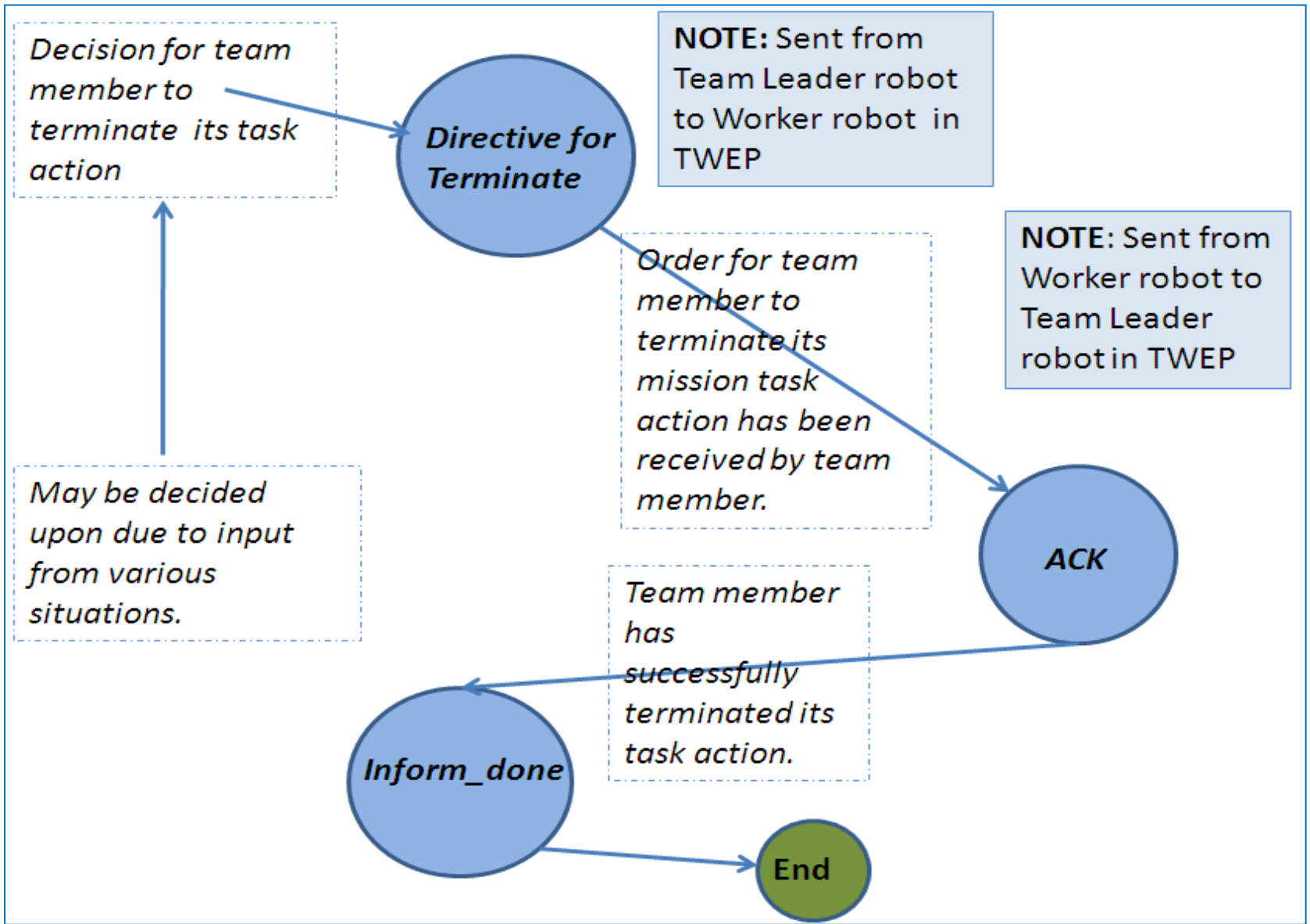
The TWE Protocol is ***central*** to the Mission activity; it provides the Context in which the other protocols are utilized.



Mini-Protocol : Worker Robots Proactively Provide Status to the Team Leader Robot

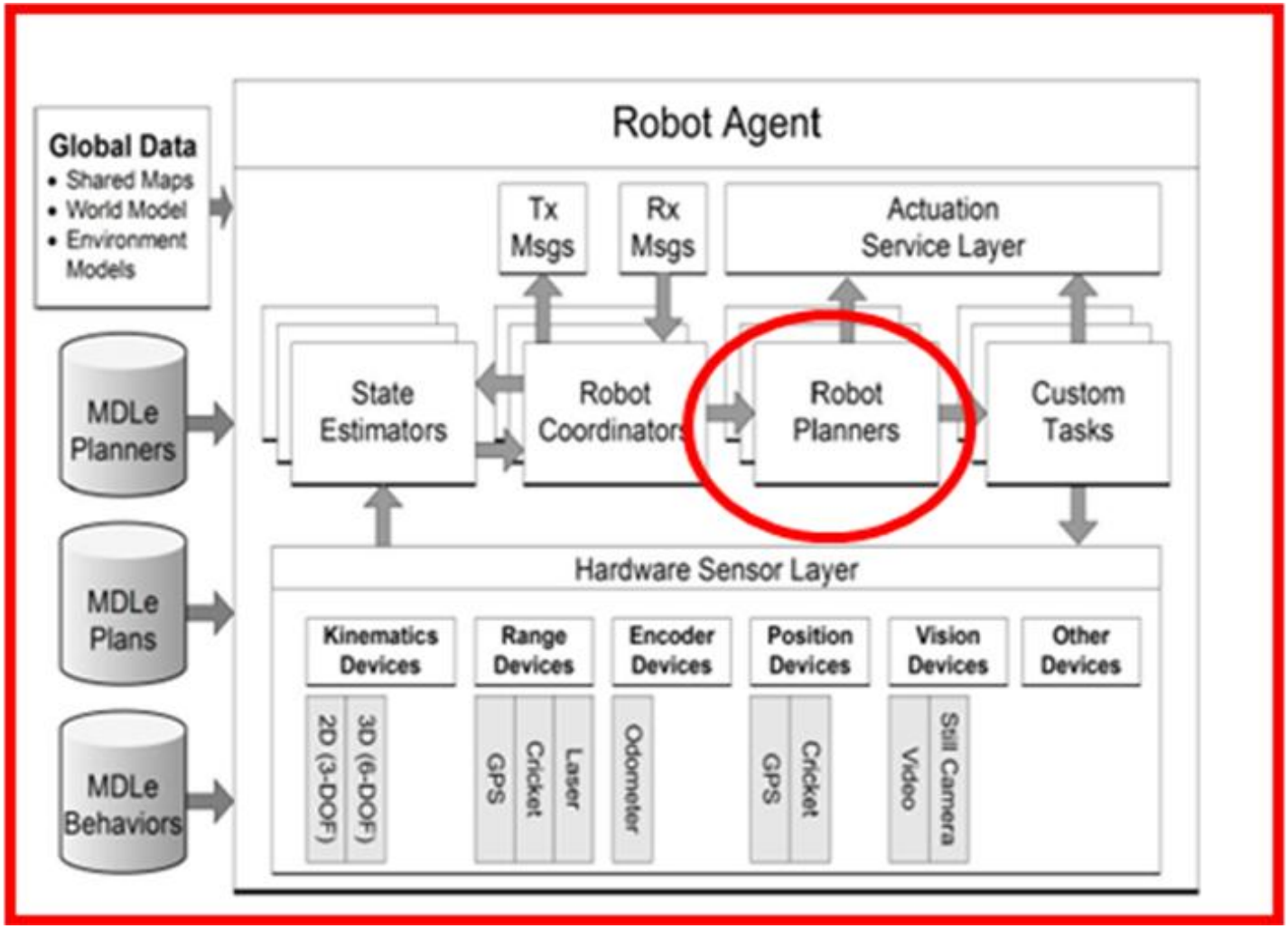


FSM Form of the Interrupt Phase: With Transition Conditions



Robot Agent: The Starting Point

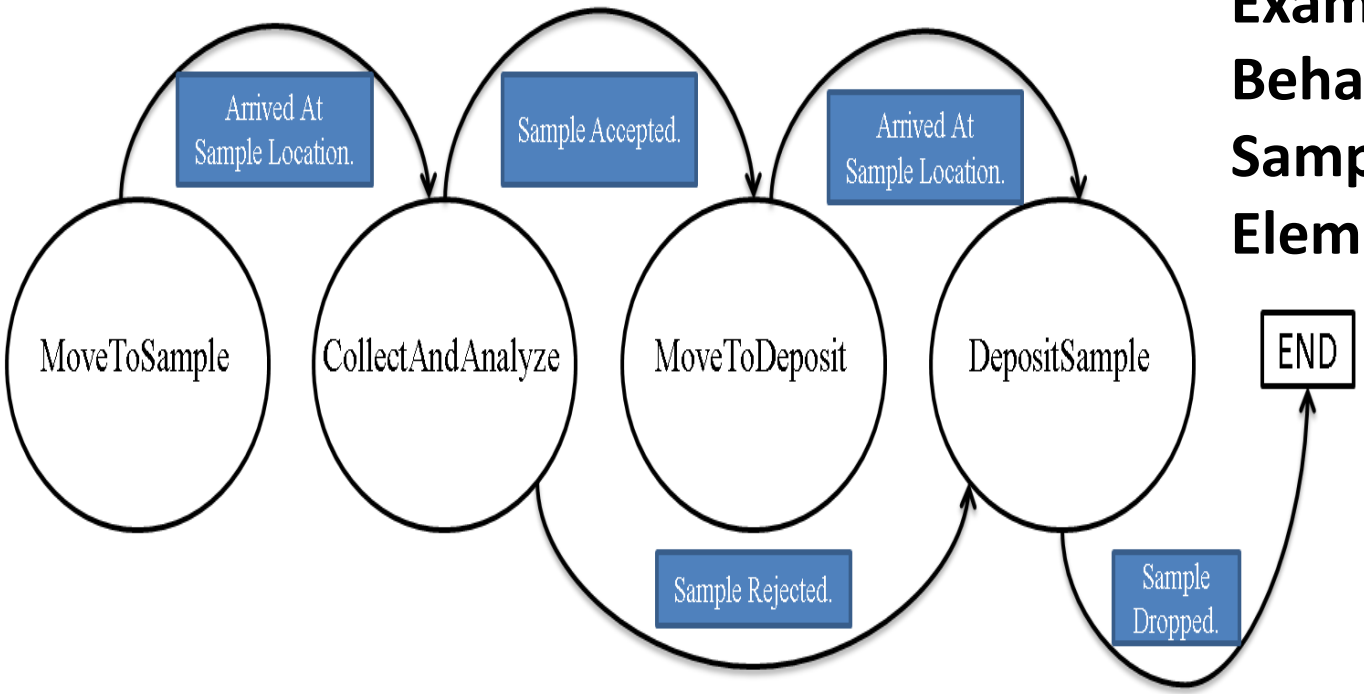
Note Message Sending and Receiving Capability



Kulis, Z. Manikonda, V. Azimi-Sadjadi, B. Ranjan, P. 2008. The Distributed Control Framework: A Software Infrastructure for Agent-Based Distributed Control and Robotics. American Control Conference, 1329-1336.

Robot Agent: The Starting Point /2

•Planning is supported by an execution engine for behaviors that are expressed in the Motion description Language, enhanced (MDLe). Tasks that are specific to a domain may be addressed through the User Task. Lower level behaviours.
•MDLe is used to build higher level (movement) behaviors from lower level atoms.



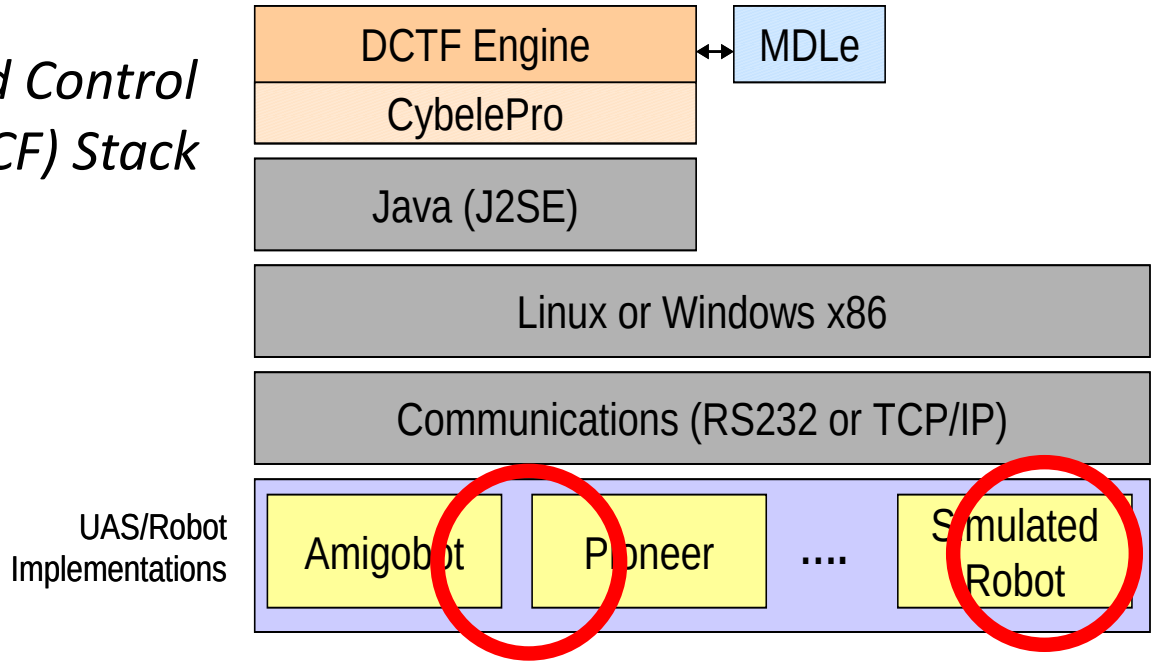
Example: Lower Level Behaviors for a Rock Sampling Mission Element.

Robot Agent: The Starting Point /3

- The Distributed Control Framework robotics framework is layered over the CybeleTM platform, which is an agent development and execution environment.
- This allows the development of the Robot agent and supports Robot agent – Robot agent communication.

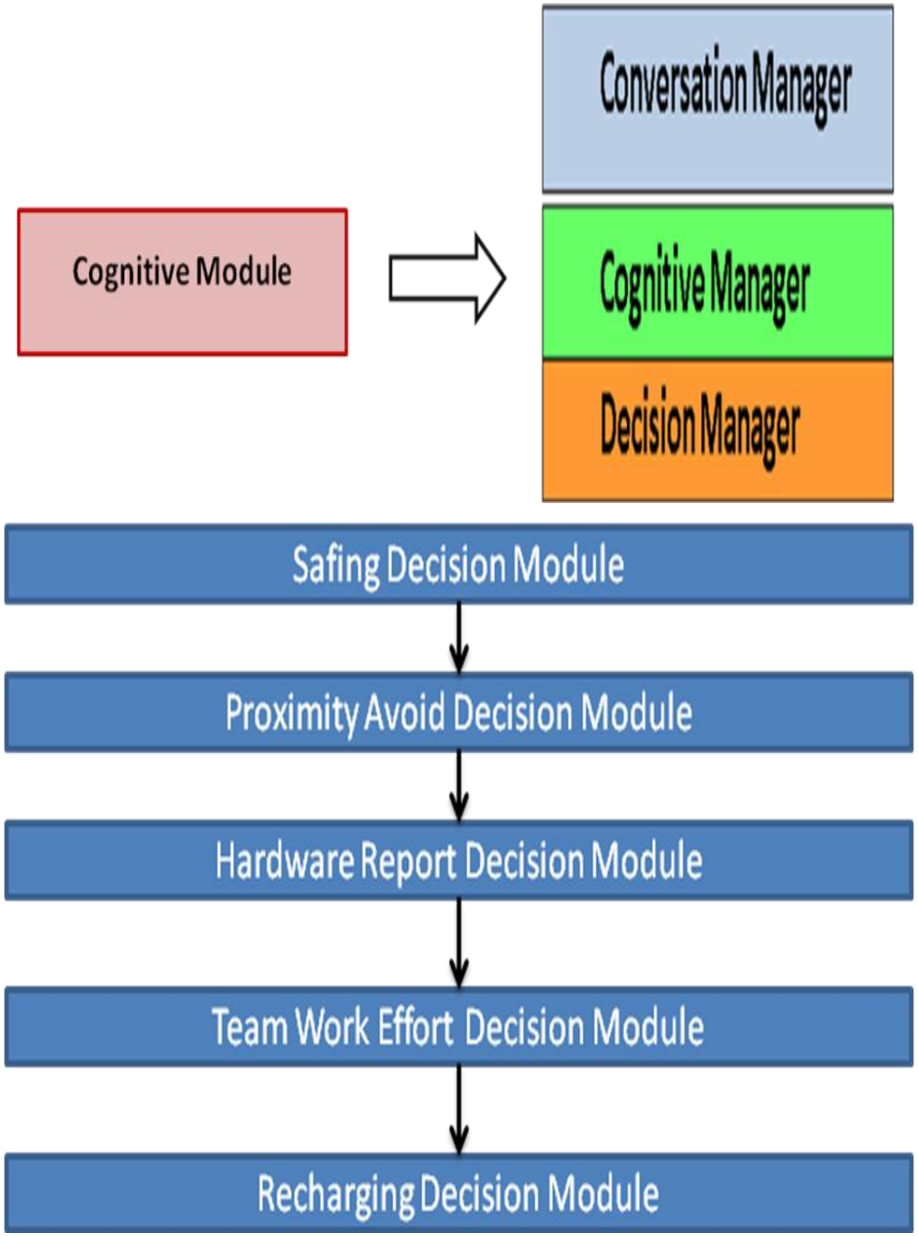
The Distributed Control Framework (DCF) Stack

The Agent platform



Protocol Aware Cognitively Enhanced Robot Agent

- The Robot Agent already supports message transmission and receiving.
- What is needed is the ability to recognize the protocol messages, to utilize information provided in them in decision making, and to respond using protocol supported messages when appropriate: Manager structure
- The Decision Manager must walk through all encoded rules. The protocol-related foci of concerns organize the rule sets into modules.
- The rule sets execute in the order of the modules, according to the implementation, which is flexible and reflects the priorities of the domain.
- Different weights can be assigned to the decision conditions, affecting action selection.



Console View: Protocol-driven Message Exchanges between Team Leader robot and Worker robot

Active Conversations

simagent.hasizeroteamlead000:0 | simagent.hasizeroteamlead000:1

Timestamp	Sender ID	Receiver ID	Protocol Name	Phase	Performative	Context
Mon Nov 17 12:07:37 EST 2008	simagent.hasizeroteamlead000	simagent.hasizeroworker001	team_work_effort	initiate_mission	query	mission_load
Mon Nov 17 12:07:38 EST 2008	simagent.hasizeroworker001	simagent.hasizeroteamlead000	team_work_effort	initiate_mission	reply	mission_load
Mon Nov 17 12:07:38 EST 2008	simagent.hasizeroteamlead000	simagent.hasizeroworker001	team_work_effort	initiate_mission	directive	mission_load
Mon Nov 17 12:07:39 EST 2008	simagent.hasizeroworker001	simagent.hasizeroteamlead000	team_work_effort	initiate_mission	ack	mission_load
Mon Nov 17 12:08:35 EST 2008	simagent.hasizeroworker001	simagent.hasizeroteamlead000	team_work_effort	initiate_mission	inform_done	mission_load
Mon Nov 17 12:08:36 EST 2008	simagent.hasizeroteamlead000	simagent.hasizeroworker001	team_work_effort	execute_mission	directive	start_task
Mon Nov 17 12:08:36 EST 2008	simagent.hasizeroworker001	simagent.hasizeroteamlead000	team_work_effort	execute_mission	ack	start_task
Mon Nov 17 12:31:50 EST 2008	simagent.hasizeroworker001	simagent.hasizeroteamlead000	team_work_effort	execute_mission	inform	status
Mon Nov 17 12:31:51 EST 2008	simagent.hasizeroteamlead000	simagent.hasizeroworker001	team_work_effort	execute_mission	ack	status
Mon Nov 17 12:31:51 EST 2008	simagent.hasizeroteamlead000	simagent.hasizeroworker001	team_work_effort	terminate_mission	directive	terminate
Mon Nov 17 12:31:52 EST 2008	simagent.hasizeroworker001	simagent.hasizeroteamlead000	team_work_effort	terminate_mission	ack	terminate
Mon Nov 17 12:31:52 EST 2008	simagent.hasizeroworker001	simagent.hasizeroteamlead000	team_work_effort	terminate_mission	inform_done	terminate

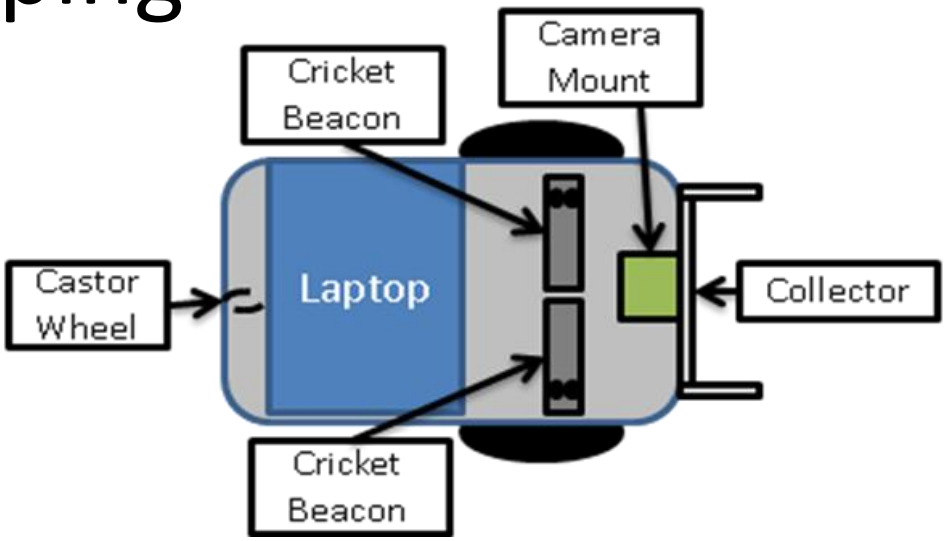
Details of protocol-governed messages exchanges for the Initiate, Execute and Terminate phases of the Team Work Effort Protocol.

Hardware Prototyping

•Additional adaptations of the Robot agent's software:
- image processing enhancements
- modifications to the lower level behaviors for rock sampling behavior.

•Rocks are 'blocks'; pink rocks are collected and green rocks are rejected.

•The catcher on the front of the robot collects the blocks.



Protocol-aware, Cognitively Enhanced Robot Agent hosted on an Amigo-bot platform; hardware includes:
- Ubuntu Linux Version 8.04 OS for Protocol Aware Cognitively Enhanced Robot agent
- Cricket beacon receivers
- 802.11 wireless on robots, with hub in lab
- Camera
- Collector

Screen Shots from Hardware Demo in Lab /1

- **Initiate phase: Team Leader queries Worker robot regarding mission load.**
  - **Message exchange but no motion yet.**
- **Initiate Phase: Worker robot proceeding to work area. Line in carpet indicates rock cart:**
  - The Team Leader has issued a ‘directive’ with a ‘mission_load’ context.
  - After acknowledging, the Worker robot’s decision structures invoke a task to ‘proceed to location’

Screen Shots from Hardware Demo in Lab /2

- Worker robot arriving at the location to start the rock sampling.
  - The Worker robot indicates its location with an ‘inform_done’ message.
  - The Team Leader can send a ‘directive’ with a ‘start_task’ and begin the Execution phase
- **All pink rocks are collected.** Team Leader terminates the mission with a ‘directive’, ‘terminate’ message

Outline

- Software Agents in Context
- Standards
- Applications: Some Examples
  - Multi-agent System for Simulations
  - Agents and Robots
- **JADE: Java Agent Development Environment**
  - **DEMO**

Resources

Agent List:
http://lists.cs.umbc.edu/mailman/listinfo/agents/

IEEE/FIPA Standards:
http://www.fipa.org/

Conference: AAMAS: Autonomous Agents and Multi-Agent Systems

Various Textbooks

Open Source JADE Software: http://jade.tilab.com/

Thank You

Any Questions ?