Towards Seamless Agent Middleware

The first question
- How are we going to implement the paradigm shift, under the heavy weight of legacy?

Mainstreaming Agent Technologies
[OR04]
- Observing the state of agent technologies nowadays
- Focussing on agent middleware
- Devising out a possible scenario
The Technology Life-Cycle

A successful technology from conception to abandon
- First ideas from research
- Premiere technology examples
- Early adopters
- Widespread adoption
- Obsolescence
- Dismissal

Often, however, this does not happen
- New technologies fail without even being tried for real
- Which are the factors determining whether a technology will either succeed or fail?
Dimensions of a Technology Shift

Technology scenario has at least three dimensions

- **Programming paradigm**
  - new technologies change the way in which systems are conceived
- **Development process**
  - new technologies change the way in which systems are developed
- **Economical environment**
  - new technologies change market equilibrium, and their success is affected by market situations

3-D space for a success / failure story

- What will determine the success / failure of agent-based technologies?
The Programming Paradigm Dimension

Pushing the paradigm shift

- Evangelists gain space on media
- Technological geeks follow soon
- Drawbacks
  - too much hype may create unsupported expectations
  - perceived incompatibility with existing approaches
  - possible dangers for conceptual integrity

Middleware for the paradigm shift

- Technology support to avoid unsupported claims
- Seamlessly situated agents vs. wrapper agents
  - communication actions towards agents
  - pragmatically actions towards objects

- This allows agents to be used in conjunction with sub-systems adopting different component models
The Development Process Dimension

Accounting for real-world software development

- Availability of development methods & tools is critical
  - No technology is to be widely adopted without a suitable methodological support
- Day-by-day developer’s needs should be accounted, too

Agent-Oriented Software Engineering Methodologies

- Adopting agent-based metaphors and abstractions to formulate new practises in software engineering
- Current state of AOSE methodologies
  - early development phases are typically well-studied
  - later phases are not, neither the tools, nor the fine-print details
Innovation has to be handled with care

- Stakeholders of new technologies may enjoy advantages of early positioning
- However, they often focus too much on *novelty* and *product*, rather than on *benefits* and *service*
  - “We are different” alone does not help much
  - software is a quite peculiar product: nearly zero marginal cost, and almost infinite production capability
Agent-Oriented Middleware & Infrastructures

- Promoting agent-oriented technologies through integration with existing object-oriented middleware & infrastructures
- Creating a no-cost space for agent technologies
- Notions like e.g. *ontology* or *coordination as a service* [VO06], which are made available to components of any sort
  - where (agent) technologies are no longer “sold” as whole packages
  - whose choice do not require any design commitment
  - where however agents represent the most effective choice for most components
- allow agent metaphors to add their value to existing systems with no assumption on the component model
Andrea Omicini and Giovanni Rimassa. 
Towards seamless agent middleware. 
Proceedings.

Mirko Viroli and Andrea Omicini. 
Coordination as a service. 
Special Issue: Best papers of FOCLASA 2002.
Evolution of Middleware: Towards Agents
Distributed Systems
Sistemi Distribuiti

Andrea Omicini
andrea.omicini@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna a Cesena

Academic Year 2014/2015