Agent Oriented Software Engineering Methodologies

Andrea Omicini

Alma Mater Studiorum – Università di Bologna
andrea.omicini@unibo.it

Panel
Agent Oriented Methodologies and Programming Languages
Towards Practical Systems
AAMAS 2009, Budapest, Hungary
13 May 2009
Outline

1. Why We Do Not Need To Worry About AOSE
2. Why We DO Need To Worry About AOSE
3. What We Should Think / Do About AOSE
4. Bibliography
Outline

1. Why We Do Not Need To Worry About AOSE

2. Why We DO Need To Worry About AOSE

3. What We Should Think / Do About AOSE

4. Bibliography
It Works

- MAS represent indeed a novel general-purpose paradigm for software development
- Agent-based computing promotes designing and developing applications in terms of
  - autonomous software entities
  - situated in an environment
  - that can flexibly achieve their goals by interacting with one another in terms of high-level protocols and languages
- These features are well suited to tackle the complexity of developing software in modern scenarios [Zambonelli and Omicini, 2004]
  - the autonomy of application components reflects the intrinsically decentralised nature of modern distributed systems, and can be considered as the natural extension to the notions of modularity and encapsulation for systems that are owned by different stakeholders
  - the flexible way in which agents operate and interact (both with each other and with the environment) is suited to the dynamic and unpredictable scenarios where software is expected to operate
  - the concept of agency provides for a unified view of artificial intelligence (AI) results and achievements, by making agents and MAS act as sound and manageable repositories of intelligent behaviours
It is the Only Thing with Some Chances to Work

- So-called “traditional” approaches to SE have already shown their limits
  - “Failure” is still a typical result of many large-scale software projects in the real world
  - Costs are clearly too high: Small-medium enterprises cannot really afford neither in-house nor external software development for their specific, pressing needs
  - In practice, time is always a problem in complex application scenarios

- Aspect-oriented programming / methodologies?
  - quite successful
  - not enough: there is not enough “aspectdness” to really face the complexity of pervasive systems, for instance
The Issue of Meta-Model

- The fact is, that only AOSE methodologies (and more generally, AO techniques) have the richness to face complexity at the right level of expressiveness
  - Agent / Society / Environment
  - Agents & Artifacts
  - Intelligent agent architectures
  - Organisational / Normative views on MAS
  - ...

- Many meta-models, still AOSE makes a uniform coherent view over complex artificial systems available
  - complex distributed systems
  - intelligent systems
  - service-oriented systems
  - robotics systems
  - pervasive systems
  - adaptive & self-* systems
  - socio-technical systems
  - ...

- Objects? Aspects? Services? ...???
  - Please, be serious
The Issue of Process

- AOSE methodologies are the only viable approach to the engineering of socio-technical systems
  - where the distinction between human and artificial actors first blurs then fades away
- AOSE researchers are running along the path of meta-modelling to build a coherent and effective model of SE processes
  - new “Design Process Documentation and Fragmentation” IEEE FIPA Working Group coming next
- The SE Process is a Socio-technical Systems
  - the AOSE paradigm is the only one capable of directly expressing both the system and the process in the same uniform way
The State of the Art

- Research on AOSE is alive
  [Henderson-Sellers and Giorgini, 2005, Bergenti et al., 2004]
  - AOSE methodologies still under development, toward mature state
  - developing lines of research on meta-modelling, fragmentation & method composition, on effective AO tools & frameworks, on the bottom-up approaches emerging from leading AO technologies...
  - the IJAOSE journal seems to have finally achieved a good standard
  - some funded projects on pure AOSE themes – e.g., “MEnSA” in Italy
  - a lot of cooperation among the AOSE researchers, in Europe in particular, but not limited to
  - AOSE TFG’s last meeting was in Bath, UK, last December, even without funding [Bernon et al., 2005]

- Teaching of AOSE is gaining momentum [Luck et al., 2004]
  - A number of MAS courses spend some time on AOSE methodologies, at least in European universities
Outline

1. Why We Do Not Need To Worry About AOSE
2. Why We DO Need To Worry About AOSE
3. What We Should Think / Do About AOSE
4. Bibliography
Issues & Problems

- AOSE methodologies are no longer perceived as a cool topic by many researchers
  → not enough research on new AOSE issues
- Tools are not perceived by many researchers as essential elements of their research
  → not enough work on effective tools
- Too few projects on the specific issues
  → not enough money invested
- Slow penetration in the industry
  → not enough practice & experience for AOSE methodologies to mature
- Teaching methodologies is boresome
  → not enough teaching for AOSE methodologies to spread fast
Outline

1. Why We Do Not Need To Worry About AOSE
2. Why We DO Need To Worry About AOSE
3. What We Should Think / Do About AOSE
4. Bibliography
What Should We Think about AOSE?

Think

- AOSE methodologies are boring but cool
  - they are the only hope for the definition of a solid engineering process for artificial systems of tomorrow
  - just think of artificial systems instead of simple software ones
- AOSE methodologies should aim at artificial systems instead of mere software ones
  - they could work as the basis for a general discipline of system engineering
- AO technologies & AOSE frameworks should be aimed at making the costs of paradigm shift as low as possible
  - they should promote coherent and well-founded blending of OO technologies and models in the MAS conceptual context
What Should We Do about AOSE?

Do

- Using AOSE methodologies in projects and complex application problems
- Bridging between “traditional” and AO software engineering techniques and notations
- Driving the development of AOSE tools within projects and advanced applications
- Teaching AOSE methodologies systematically in SE courses, linked to non-trivial student projects
Conclusion

So What?

- Keep on researching in AOSE – so much still to do & achieve
- Do not fret about AOSE – so much time still ahead of us
- Look back carefully – most of what we learned about SE is likely to be of some use in AOSE
- Look forward with an open mind – most of the systems to be engineered in the near future have no answer from traditional SE theory and practice, new ideas required
- But mostly, take care of AO technologies, which will be the key to the widespread practice for AO-whatever
Outline

1. Why We Do Not Need To Worry About AOSE
2. Why We DO Need To Worry About AOSE
3. What We Should Think / Do About AOSE
4. Bibliography
Bibliography I


Bibliography II


Agent Oriented Software Engineering Methodologies

Andrea Omicini

Alma Mater Studiorum – Università di Bologna
andrea.omicini@unibo.it

Panel
Agent Oriented Methodologies and Programming Languages
Towards Practical Systems
AAMAS 2009, Budapest, Hungary
13 May 2009