Advancing Object-Oriented Standards Toward Agent-Oriented Methodologies: SPEM 2.0 on SODA

Ambra Molesini Elena Nardini Enrico Denti Andrea Omicini

Alma Mater Studiorum – Università di Bologna
{ambra.molesini,elena.nardini,enrico.denti,andrea.omicini}@unibo.it

Dagli Oggetti agli Agenti
Evoluzione dell'agent development: metodologie, tool, piattaforme e linguaggi
Palermo, Italy
17th November 2008

Outline

Scenario

- The creation of a single universally-applicable methodology is a recognised *chimera*
  - → Software designers tend to define their own problem-specific methodology by means of the *Method engineering* technique
  - → New methodologies can be created starting from existing methodology parts (method fragments)
- A unified meta-model is needed, allowing existing methodologies to be represented in a uniform way
- The Software Process Engineering Metamodel (SPEM) 2.0 [Object Management Group, 2008] is an OMG standard and seems to be the natural candidate as the *process meta-model*

Objectives

→ Exploring SPEM 2.0 applicability to the AOSE domain, whose abstractions and mechanisms are particulary suited to the design and development of complex software systems

→ Comparing the meta-modelling powers of SPEM 2.0 and SPEM 1.0

- SODA (*Societies in Open and Distributed Agent spaces*) is a significant case study for stressing SPEM's strengths and weaknesses
  - it is an AOSE methodology
  - it focusses on modelling the *social issues* and the application *environment*
  - it exploits mechanisms for capturing the *layered structure* of complex systems

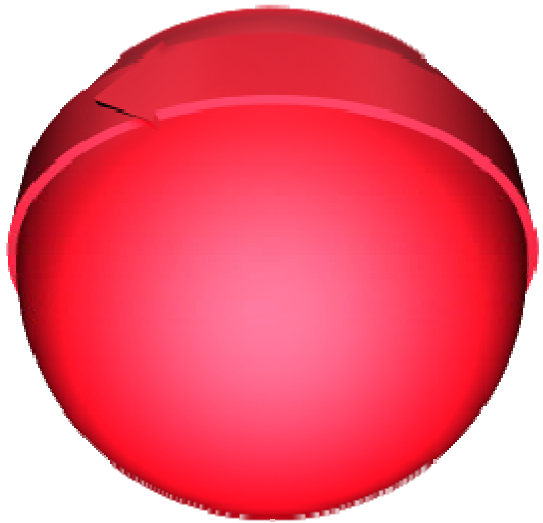
Outline

Outline

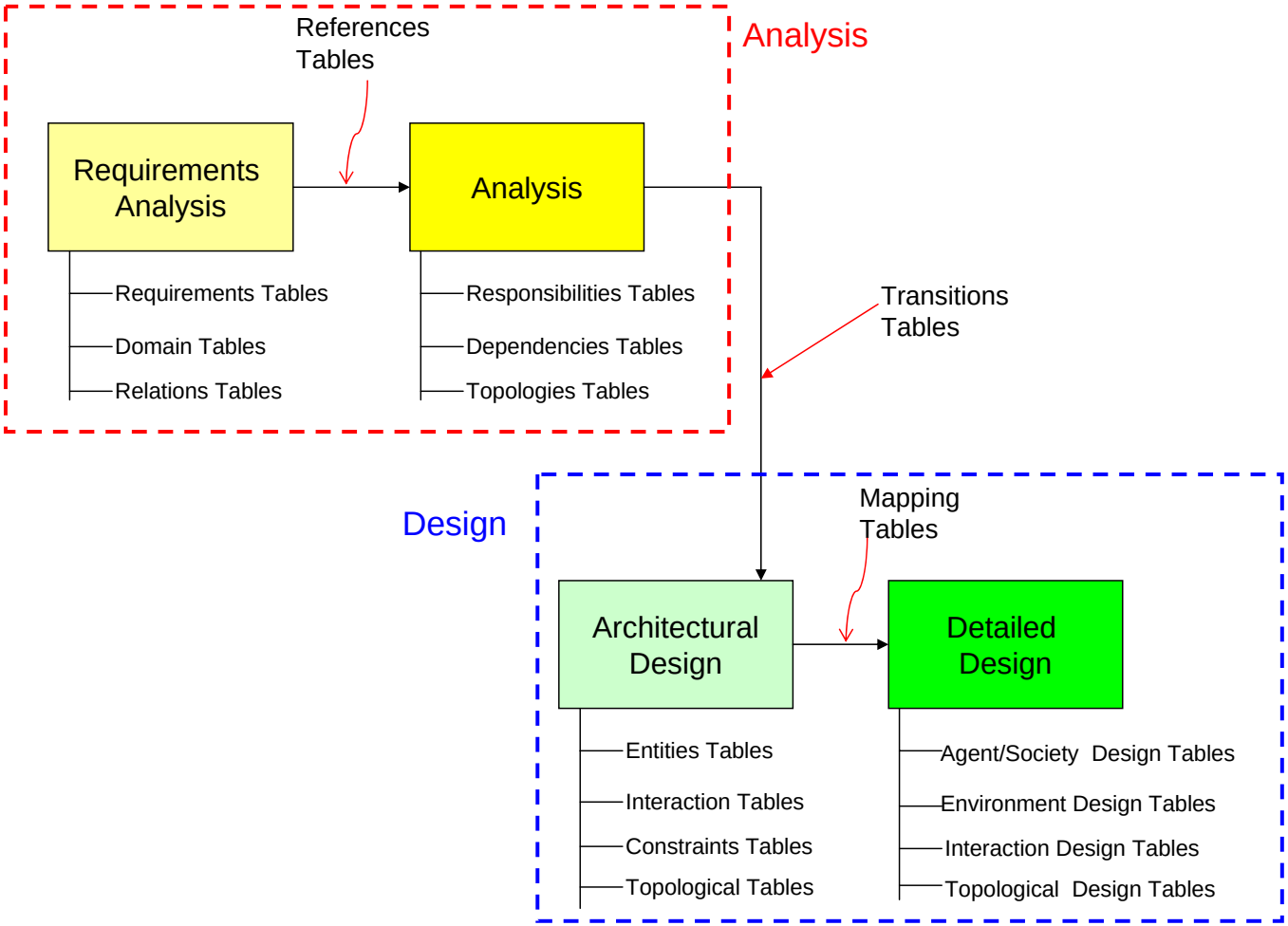
SODA: Societies in Open and Distributed Agent spaces

SODA ...

- ... is an agent-oriented methodology for the analysis and design of agent-based systems
- ... focuses on inter-agent issues, like the engineering of societies and environment for MAS [Omicini, 2001]
- ... adopts agents and artifacts – after the A&A meta-model – as the main building blocks for MAS development [Molesini et al., 2005]
- ... introduces a simple *layering* principle in order to cope with the complexity of system description [Molesini et al., 2006b]
- ... adopts a tabular representation

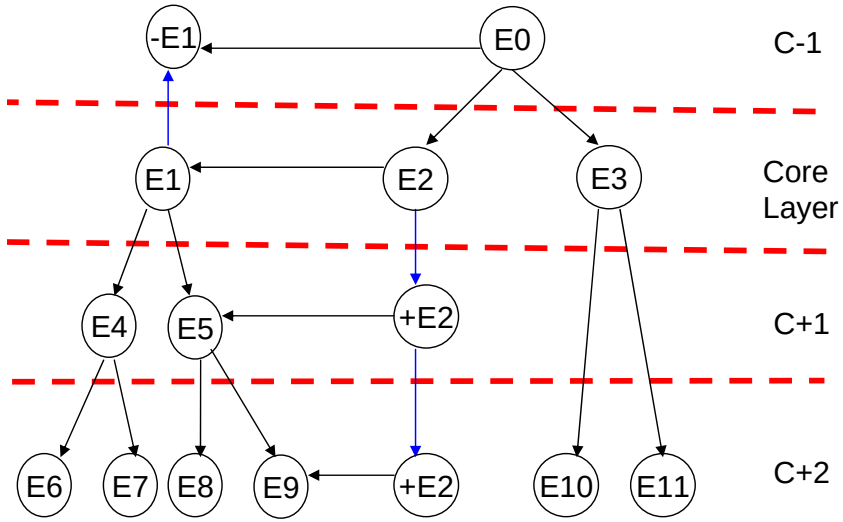


SODA: An Overview



SODA: The Layering Principle

- The layering principle is achieved by means of the *zoom* and *projection* mechanisms
- Two kinds of zoom
  - *in-zoom*: from abstract to a more detailed layer
  - *out-zoom*: from detailed to a more abstract layer
- The *projection mechanism* projects entities from one to another layer



Outline

SPEM 2.0 as Process Meta-model

- In order to obtain an effective representation of processes, a specific set of concepts and symbols is needed
- UML *profiles* can be used

  a UML *profile* is an UML extensions for building UML models related to specific domains
- SPEM is a UML profile which extends UML expressiveness towards the modelling of software development processes
  - Overcome the limits of UML by adding all the concepts and symbols required to represent a software development process
  - Provides users with an easy-to-understand notation
  - A wide community of software developers is familiar with UML

    → A large community can reuse its knowledge in the software process domain

SPEM 2.0

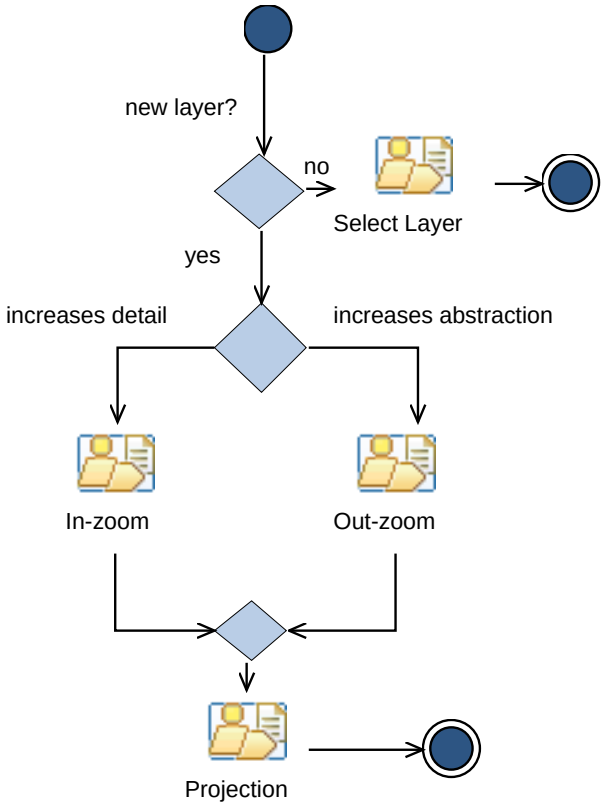
- SPEM is based on the idea that a software development process is a collaboration between active abstract entities called *roles* which perform operations called *activities* on *work products*
- The goals of SPEM 2.0 are to:
  - support the representation of one specific development process
  - support the maintenance of several unrelated processes
  - promote process reusability by means of Capability patterns
  - separate
    - ★ Method Contents — introduce the concepts to document and manage development processes through natural language description
    - ★ Processes — defines a process model as a breakdown or decomposition of nested *Activities*, with the related *Roles* and input / output *Work Products*

Outline

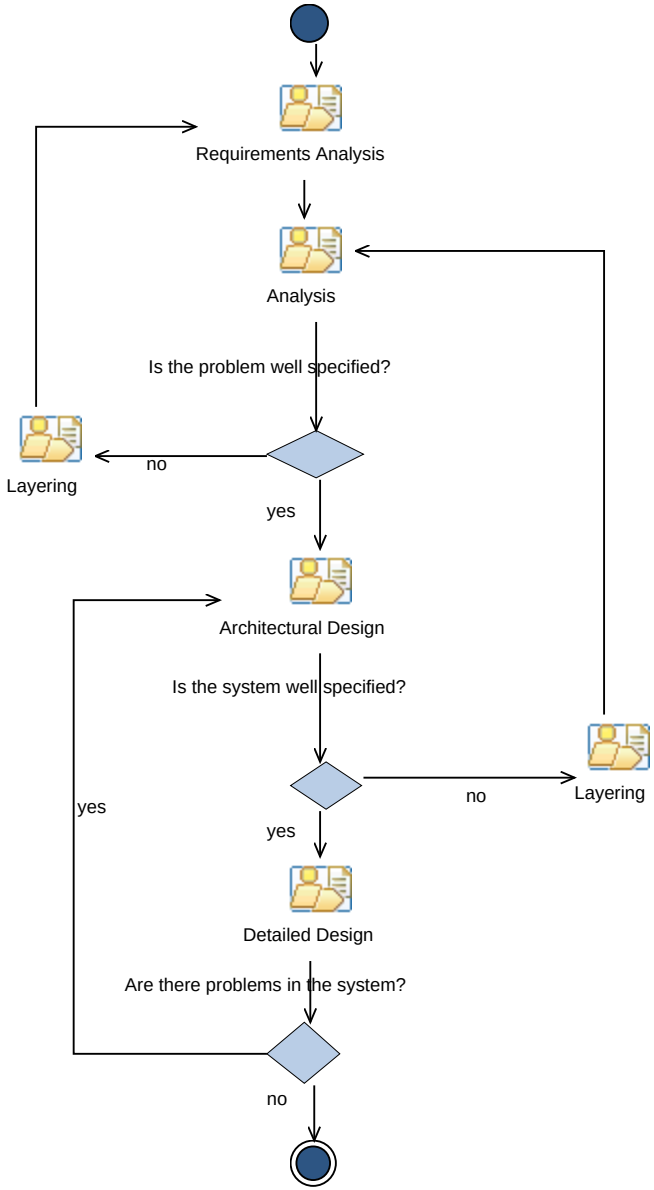
SODA Process Organisation

- Four sub-phases: Requirements Analysis, Analysis, Architectural Design, Detailed Design
- Each sub-phase is modelled as a separate and independent Method Content
- A specific process is defined for each sub-phase
- The specific processes are re-used to create the whole SODA process
- Each sub-phase – and each model in the sub-processes – is represented as an activity, related to the corresponding SPEM's Task in the specific Method Content
- The Layering Capability pattern is adopted

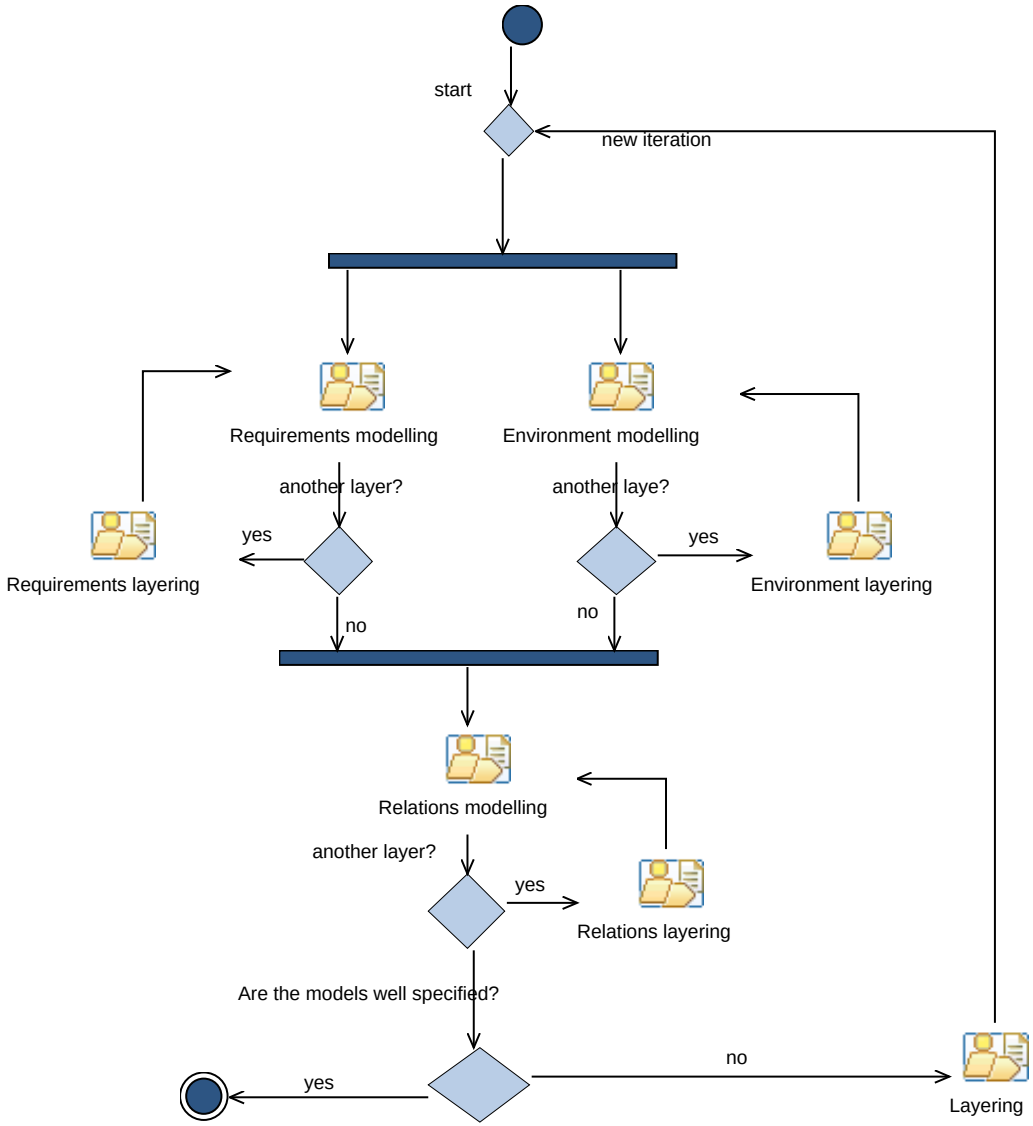
Layering in SODA: The Process



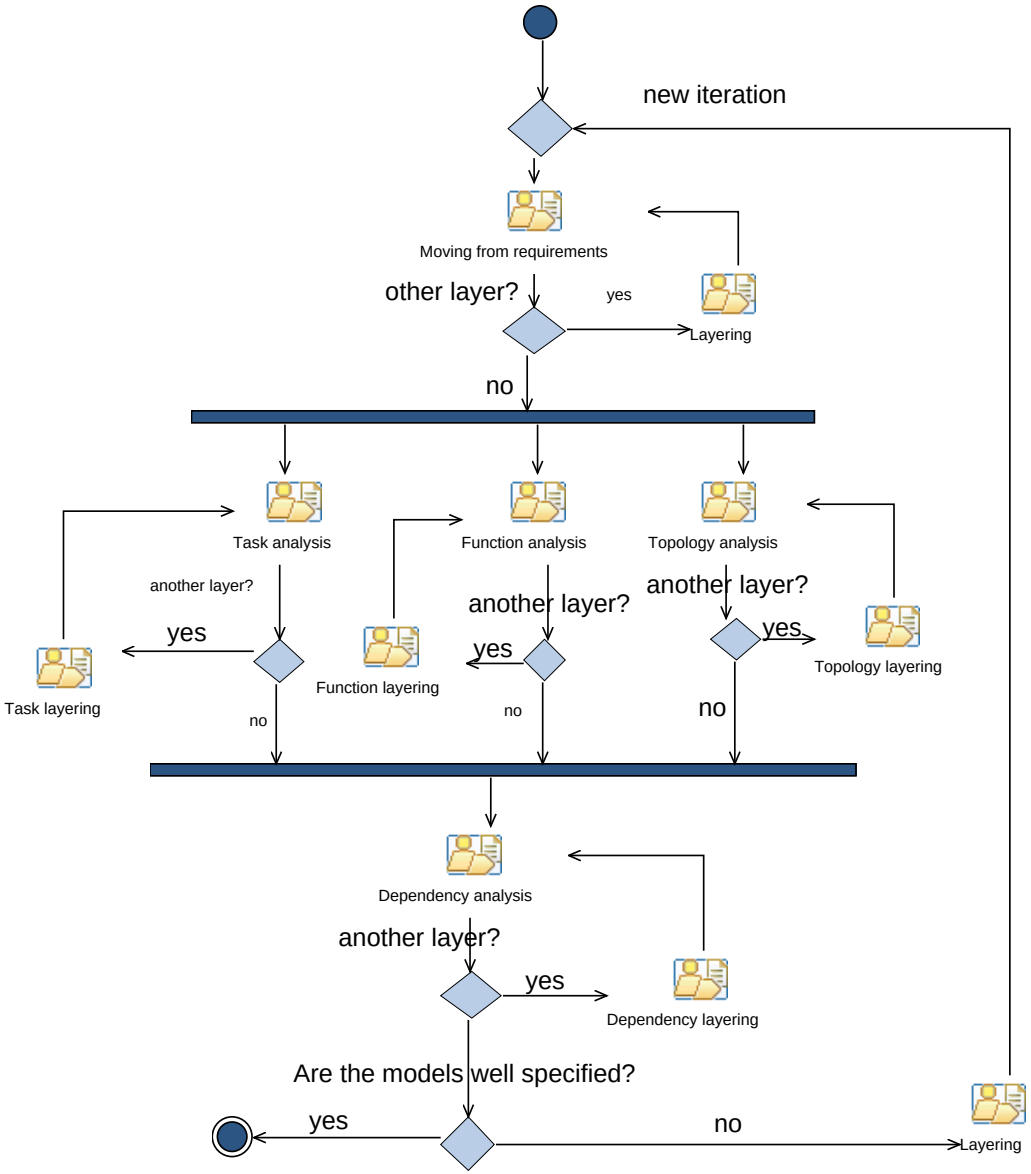
The SODA Process



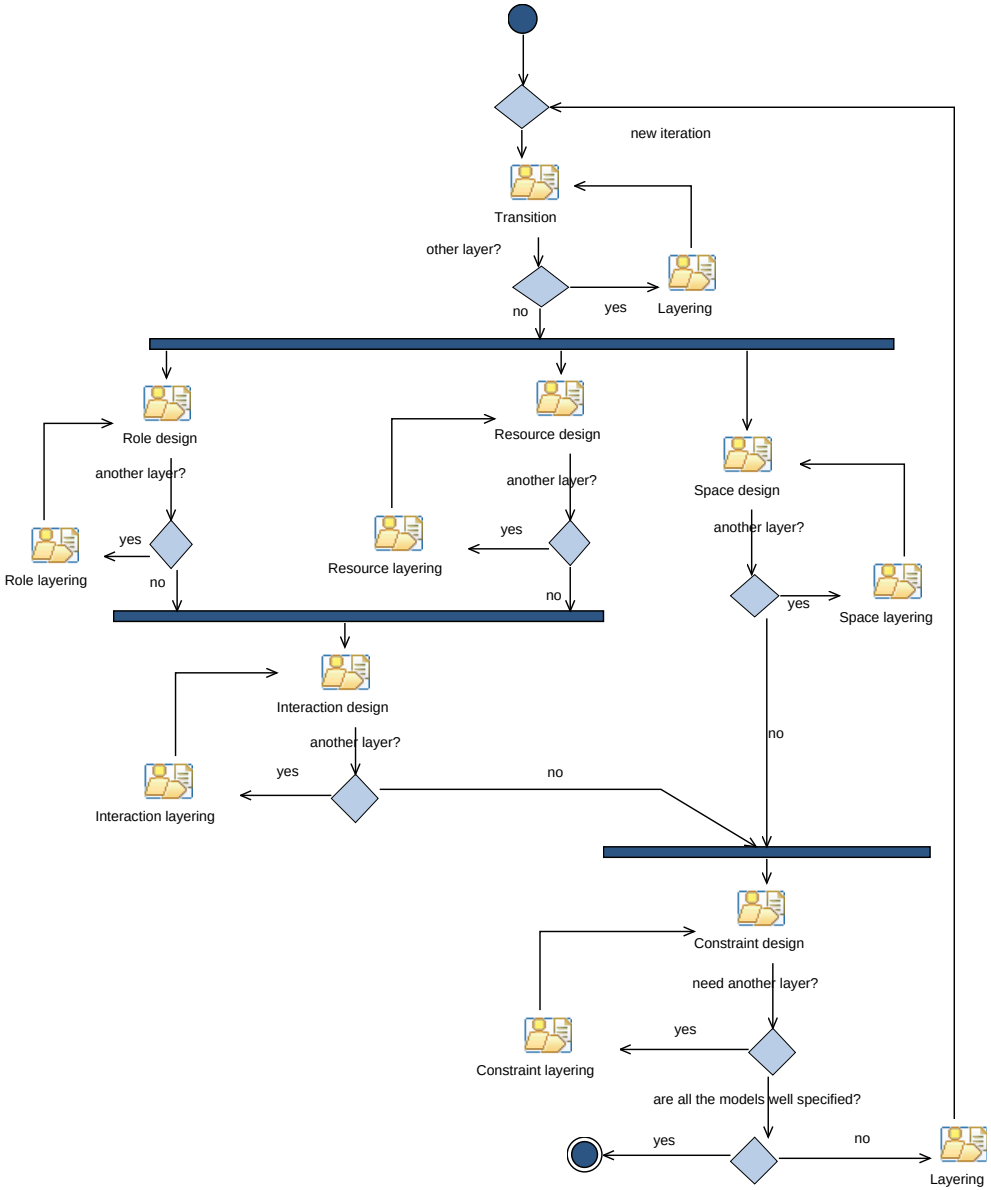
Requirements Analysis Process



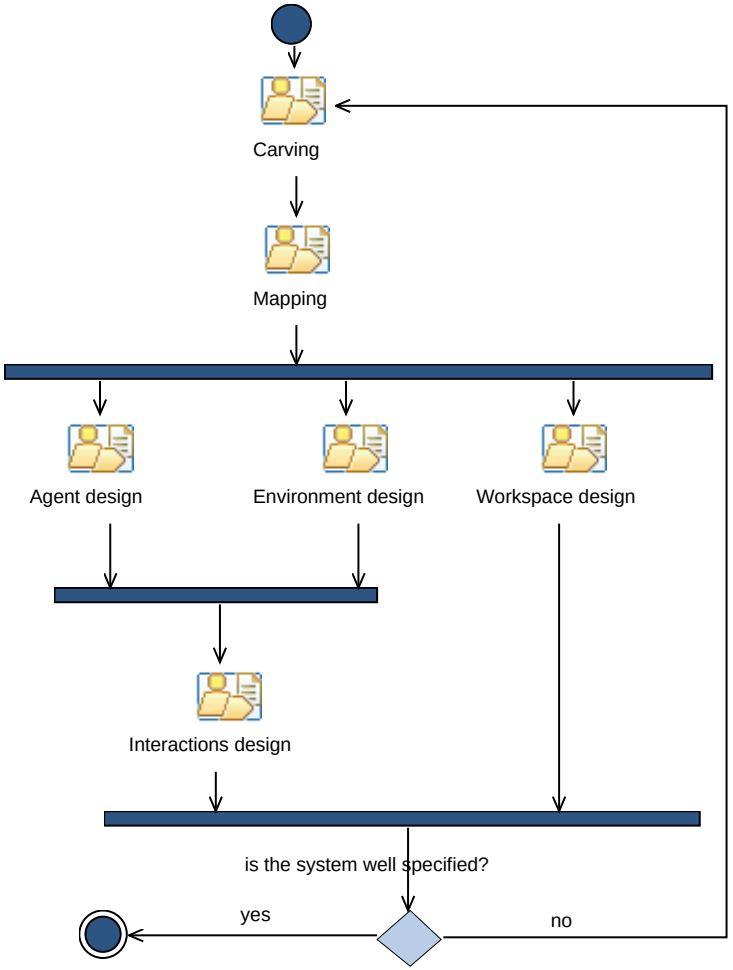
Analysis Process



Architectural Design Process



Detailed Design Process



Outline

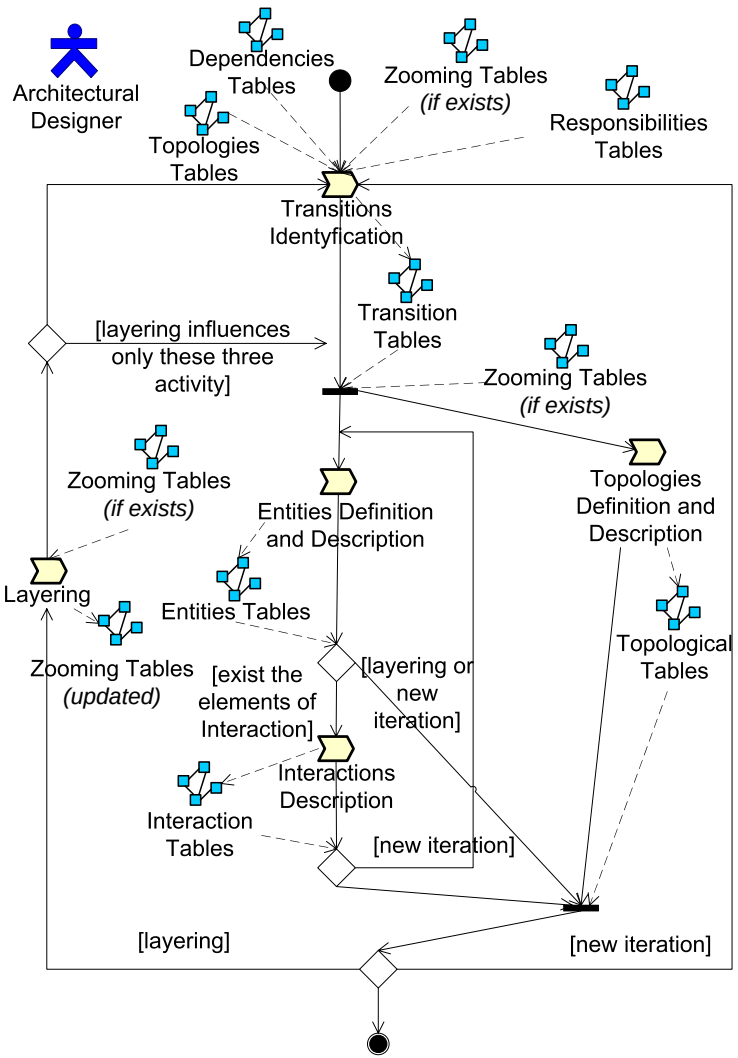
SPEM 1.0 vs. SPEM 2.0 in an AO Context: General Experience

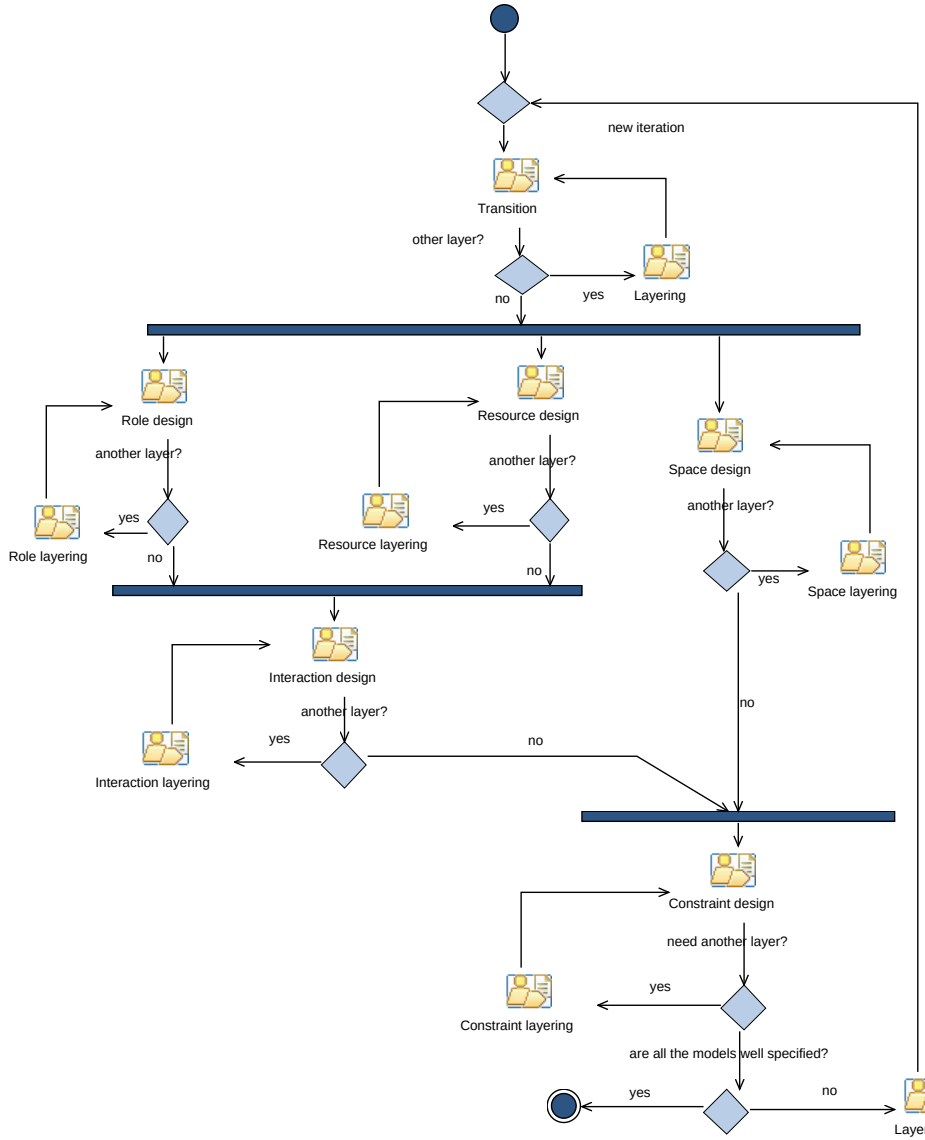
- Despite its origin in the object-oriented context, SPEM 2.0 could be applied to the agent-oriented SODA process quite naturally
- The limits in expressiveness and readability of SPEM 1.0 [Nardini et al., 2008] are mostly addressed by the new version of SPEM
- The software development process and its phases are similar in any methodology
- However, agent-oriented methodologies introduce a rich set of abstractions and mechanisms
- This sometimes have stressed SPEM 1.0 to its limits, showing weakness in facing the increasing complexity
- In particular, UML diagrams often become nearly unreadable when applied to AOSE methodologies

SPEM 1.0: Limits in Expressiveness

1. Activity Diagrams and abstractions did not easily capture the SODA layering principle: layering is represented as a simply activity and there is no way to detail the layering sub-process without reporting in the Activity Diagram all the layering sub-activities
2. WorkProduct elements are characterised by a unique symbol, which makes it difficult to model the state changes of a WorkProduct during the process evolution
3. UML Diagrams often become unreadable due to the too many elements required to represent a process

SPEM 1.0 vs. SPEM 2.0: the Architectural Design Process





SPEM 1.0 vs. SPEM 2.0: Layering

- SPEM 2.0 addresses the layering representation issue by providing the *capability pattern* mechanism that makes it possible to represent a process pattern as a single activity, hiding its internal structure
- Such a pattern allows engineers to realise more understandable and readable diagrams by hiding the process complexity behind the Activity abstraction

→ So, the different activities composing the Layering can now be detailed without reporting them in the Activity Diagrams each time, leading to a great simplification

SPEM 1.0 vs. SPEM 2.0: Workproduct

- SPEM 2.0 addresses the workproduct representation issue by extending
  - UML Activity Diagrams so as to represent the input and output parameters of an Activity
  - UML State Diagrams so as to annotate the State elements
- Such extensions enable UML State Diagrams to model the lifecycle of each WorkProduct, and relate each State element to the corresponding Activity that causes the state change

→ This makes it unnecessary to represent the Workproducts inside the Activities Diagrams as it was in SPEM 1.0

SPEM 1.0 vs. SPEM 2.0: UML Diagrams

- The issue related to the UML diagrams is already partially addressed by capability patterns that simplify the Diagrams structure
- SPEM 2.0 introduces the concept of *process reusability* and allows *Method Contents* to be defined independently of their application in the development lifecycle
- So, Method Contents can be re-used by relating their elements into a customised process
- Each UML Diagram
  - is now more readable
  - focusses only on a given portion of the Method Content / Process
  - does not contain all the "unusable" entities which are not related to the considered portion of the meta-model

→ We defined a Method Content for each SODA stage, relating them to the corresponding processes

Outline

Conclusion

- Moving from a previous work [Nardini et al., 2008] where the SODA process was modelled in SPEM 1.0, we explored here whether SPEM 2.0 addressed the weaknesses and limits of expressiveness that had clearly emerged
  - the readability of UML diagrams
  - the intrinsic complexity of Agent-Oriented methodologies
  - the lack of suitable ad-hoc entities
- Our experience indicates that SPEM 2.0 addresses such limits, by
  - introducing a clear separation between Method Contents and Processes
  - adding capability patterns
  - making it possible to express the ties between the Workproducts'states and the Activities that produce the changes in the Workproducts'states
- Summing up, SPEM 2.0 seems to overcome the major limits of its previous version, providing the right abstractions and mechanisms to model articulated process like SODA's, perhaps finding its way in the AOSE context.

Future Work

- Test SPEM 2.0 in other contexts, such as modelling the MAS infrastructures processes...
- ... with the purpose of integrating AOSE methodologies and MAS infrastructures according to the Situational Method Engineering technique [Cossentino et al., 2008].

Outline

Bibliography I


Brinkkemper, S. (1996).
Method engineering: engineering of information systems development methods and tools.
Information & Software Technology, 38(4):275–280.


Cossentino, M., Gaglio, S., Gaud, N., Hilaire, V., Koukam, A., and Seidita, V. (2008).
A MAS metamodel-driven approach to process composition.
In Luck, M. and Gómez-Sanz, J., editors, *9th International Workshop on Agent Oriented Software Engineering (AOSE'08)*, AAMAS 2009, Estoril, Portugal.


Ghezzi, C., Jazayeri, M., and Mandrioli, D. (2002).
Foundamental of Software Engineering.
Prentice Hall, second edition.

Bibliography II

Molesini, A., Denti, E., and Omicini, A. (2005).
MAS meta-models on test: UML vs. OPM in the SODA case study.
In Pěchouček, M., Petta, P., and Varga, L. Z., editors, *Multi-Agent Systems and Applications IV*, volume 3690 of *LNAI*, pages 163–172. Springer.
4th Inter. Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'05), Budapest, Hungary, 15–17 September 2005, Proceedings.

Molesini, A., Omicini, A., Denti, E., and Ricci, A. (2006a).
SODA: A roadmap to artefacts.
In Dikenelli, O., Gleizes, M.-P., and Ricci, A., editors, *Engineering Societies in the Agents World VI*, volume 3963 of *LNAI*, pages 49–62. Springer.
6th Inter. Workshop (ESAW 2005), Kuşadası, Aydın, Turkey, 26–28 October 2005. Revised Paper.

Bibliography III

Molesini, A., Omicini, A., Ricci, A., and Denti, E. (2006b).
Zooming multi-agent systems.
In Müller, J. P. and Zambonelli, F., editors, *Agent-Oriented Software Engineering VI*, volume 3950 of *LNCS*, pages 81–93. Springer.
6th Inter. Workshop (AOSE 2005), Utrecht, The Netherlands, 25–26 July 2005. Revised and Invited Papers.

Nardini, E., Molesini, A., Omicini, A., and Denti, E. (2008).
SPEM on test: the SODA case study.
In Wainwright, R. L., Haddad, H. M., Menezes, R., and Viroli, M., editors, *23th ACM Symposium on Applied Computing (SAC 2008)*, volume 1, pages 700–706, Fortaleza, Ceará, Brazil. ACM.
Special Track on Software Engineering.

Bibliography IV

Object Management Group (2008).
Software & Systems Process Engineering Meta-Model Specification 2.0.
http://www.omg.org/spec/SPEM/2.0/PDF.

Omicini, A. (2001).
SODA: Societies and infrastructures in the analysis and design of agent-based systems.
In Ciancarini, P. and Wooldridge, M. J., editors, *Agent-Oriented Software Engineering*, volume 1957 of *LNCS*, pages 185–193. Springer.
1st Inter. Workshop (AOSE 2000), Limerick, Ireland, 10 June 2000. Revised Papers.

Bibliography V


Omicini, A., Ricci, A., and Viroli, M. (2006).
Agens Faber: Toward a theory of artefacts for MAS.
Electronic Notes in Theoretical Computer Sciences, 150(3):21–36.
1st International Workshop "Coordination and Organization" (CoOrg 2005), COORDINATION 2005, Namur, Belgium, 22 April 2005. Proceedings.

Advancing Object-Oriented Standards Toward Agent-Oriented Methodologies: SPEM 2.0 on SODA

Ambra Molesini Elena Nardini Enrico Denti Andrea Omicini

Alma Mater Studiorum – Università di Bologna
{ambra.molesini,elena.nardini,enrico.denti,andrea.omicini}@unibo.it

Dagli Oggetti agli Agenti
Evoluzione dell'agent development: metodologie, tool, piattaforme e linguaggi
Palermo, Italy
17th November 2008