Artefacts in Agent-Oriented Software Engineering

Ambra Molesini

ambra.molesini@unibo.it

DEIS, Università di Bologna
Italy

Joint research with prof. Andrea Omicini, prof. Alessandro Ricci and prof. Enrico Denti

---

Outline

- Agents & Artefacts

- SODA+artefacts

- SODA+zoom

- Zooming Artefacts

- Conclusions and Future Work

1

Introduction

- Agents never live alone
  - ➢ they coexist with other agents in a MAS
  - ➢ within an *environment* where they act and interact

- The agent abstraction alone is not enough to fully model the environment

- The environmental entities are not agents

⇩

ARTEFACTS

Artefacts

- Artefacts take the form of objects or tools that agents *share* and *use* to

  - ➢ support their activities

  - ➢ achieve their objectives

- Artefacts are explicitly designed to provide some *functions* which guide their use.

- An artefact can have *responsibilities*

Example

- Coordination Artefacts
  - govern social activities
  - enable and mediate agent interaction
  - mediate the interaction between individual agents and their environment
  - capture, express and embody the parts of the environment that support agents' activities

Features & Classification

- An artefact exposes:
  - usage interface
  - operating instructions
  - function description
- Other interesting artefact features are:
  - inspectability
  - malleability
  - linkability
- A possible classification
  - individual artefact
  - social artefact
  - resource artefact

Agents & Artefacts

- Artefacts constitute the basic building block both for

  ➢ MAS analysis/modelling

  ➢ MAS development

- Agents & Artefacts can be assumed as two fundamental abstractions for modelling MAS structure

  ➢ agents speaking with other agents

  ➢ agents using artefacts in order to achieve their objectives

The New Meta-model

- Agents & Artefacts lead to new *ontological meta-model* for MASs

- Artefacts allow to

  ➢ model the environment as a first-class entity

  ➢ engineer the space of interaction among agents
    (not only mere conversations between agents, but complex agent interaction patterns)

  ➢ enrich MAS design with social/organisational structure, topological models, as well as (complex) security models

New Meta-model in AOSE

- AOSE methodologies should then enable and promote the design of artefacts

- However, no known agent-oriented methodology adopts agents and artefacts as its basic abstractions

- As a consequence, we choose to extend an existing agent-oriented methodology (SODA) with the notion of artefact
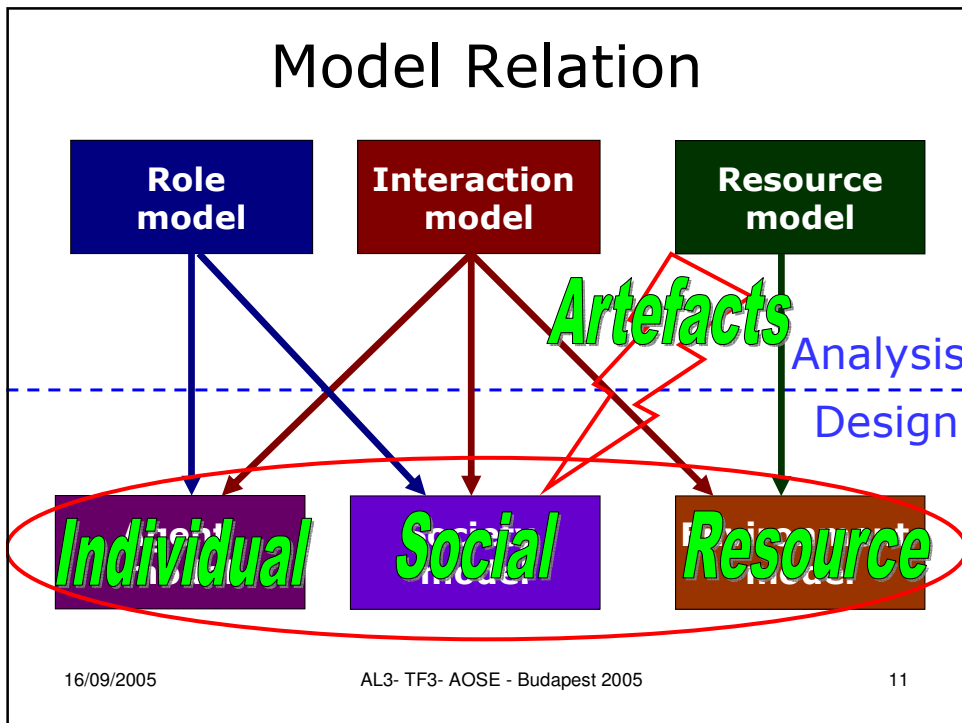
Artefacted SODA

- SODA concentrates on *inter-agent* issues, like the engineering of societies and infrastructure

- The SODA *analysis* phase is based on three models:
  - the *role model*
  - the *resource model*
  - the *interaction model*

- The SODA design phase is also based on three models:
  - the *agent model*
  - the *society model*
  - the *environment model*

Model Relation

Role model	**Interaction model**	**Resource model**

Artefacts

Analysis

Design

Individual *Social* *Resource*

Analysis

Role Table

Role	Task	Interaction Protocols
name	*task*	*list of protocols*

Group Table

Group	Social Task	Interaction Rules
name	*task*	*list of rules*

Resource Table

Abstract Resource	Service	Policy	Interaction Protocols
name	*name*	*list of permissions*	*list of protocols*

Interaction Protocols Table

Interaction Protocol	Information Required	Information provided
name	*detail of info*	*detail of info*

Interaction Rules Table

Interaction Rule	Description
Name	*detail of rule*

Design

Agent Table

Agent	Role	Individual Artefact	Interaction Protocols	Resources	Permissions
name	*roles*	*name*	*list of protocols*	*list of resources*	*list of permissions*

Society Table

Society	Group	Social Artefact	Resources	Social Rules
name	*group*	*name of artifact*	*resources*	*list of rules*

Environment Table

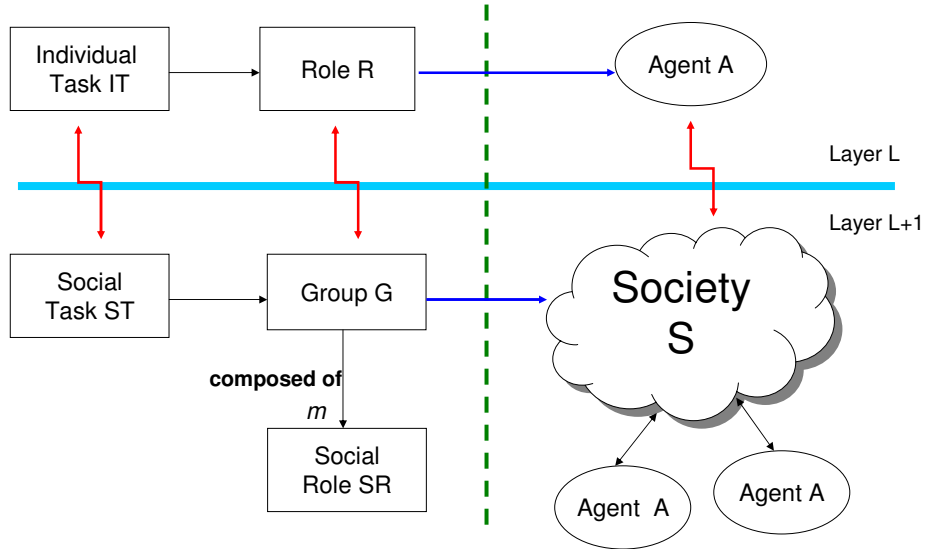| Resource Artefact	Social Artefact	Abstract Resource	Topological Abstraction	Policy	Interaction Protocols
Name	*name*	*name*	*topology*	*list of permissions*	*list of protocols*

Zooming SODA

- Goal: scaling with the complexity of task description by introducing a simple layering principle into the models of the analysis phase (role, resources, interaction models).

- Each layer contains a description of the models at a given abstraction level

- The models of the design phase become layered indirectly, since they map the layers described in the analysis phase

Zooming: The Idea

| Individual Task IT | → | Role R | → | Agent A (ellipse) |

Layer L

Layer L+1

| Social Task ST | → | Group G | → | Society S (cloud) |

Group G —
composed of
m
→ Social Role SR

Society S → Agent A (ellipse)
Society S → Agent A (ellipse)
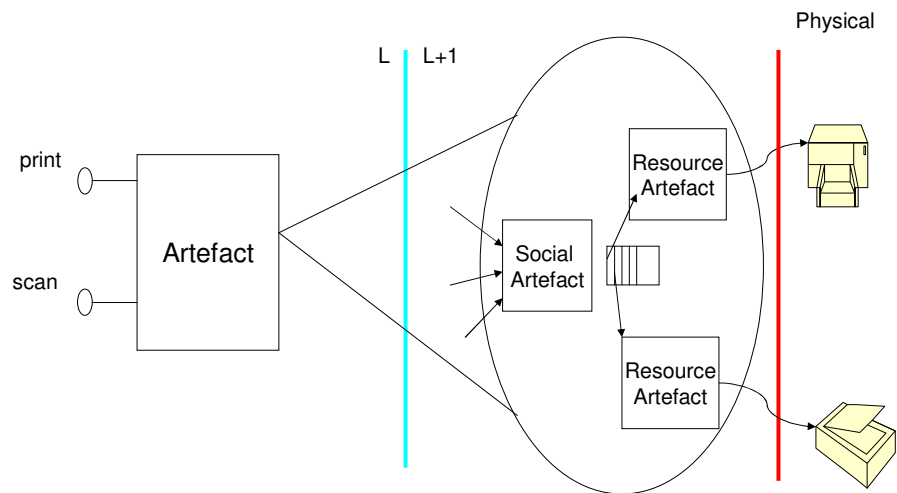

Zooming Artefacts

- Artefacts have a relevant impact on any AOSE methodology

- The notion of artefact itself is affected by the principles of the methodology where it is introduced.

- This is evident when applying zooming principle to artefacts.

Zooming Artefacts: an Example

Conclusions

- We investigate the impact of assuming a new ontological meta-model in AOSE: Agents & Artefacts

- We extend SODA with the concept of artefacts:
  - ➤ artefacts have a relevant impact on SODA
  - ➤ the SODA principle has an impact on artefacts too

- Artefacts seem to fit well with SODA models

Future Work

- Understanding the implications of the new meta-model in terms of the fundamental agent-oriented abstractions.

- Developing suitable design tools for modelling systems with SODA

- Forthcoming in SODA+zoom :

  - ➢ the *topological model*

  - ➢ the *organisational model*