Agents and Artifacts: The A&A Meta-model for Multiagent Systems

Multiagent Systems LS
Sistemi Multiagente LS

Andrea Omicini
andrea.omicini@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2007/2008

Out of the Mess

Many different & diverging definition for the notion of agent around

- Typically, a list of not well-defined properties
- "Definitory" properties are often indistinguishable from desirable ones
- Orthogonality between defining features is not even considered

How should one choose / build a definition?

- We should first make clear what are the required / desirable properties of a definition
- Only after, try to define our entities

What is a definition?

From Wikipedia

- A definition is a form of words (*definiens*) which states the meaning of a term (*definiendum*)
- Definition by *genus* and *differentia*

  genus: (the family) of things to which the defined thing belongs

  differentia: the features that distinguish the defined thing from other things of the same family

Rules for definition by genus and differentia

- A definition must set out the essential attributes of the thing defined
- Definitions should avoid circularity
- The definition must not be too wide or too narrow
- The definition must not be obscure
- A definition should not be negative where it can be positive

Explanation in the Sciences of Nature

Occam's Razor

- The explanation of any phenomenon should make as few assumptions as possible, eliminating, or "shaving off," those that make no difference in the observable predictions of the explanatory hypothesis or theory
- In short, when given two equally valid explanations for a phenomenon, one should embrace the less complicated formulation
- When multiple competing theories have equal predictive powers, one should select the one introducing the fewest assumptions and postulating the fewest hypothetical entities

Lex Parsimoniae

Entia non sunt multiplicanda praeter necessitatem
(entities should not be multiplied beyond necessity)

Definition in the Sciences of Artificial

Explanation vs. definition

- In the sciences of nature, *phenomena* are just to be observed, described, and possibly predicted, and *noumena* to be possibly understood
  - definition is just a premise to theory and explanation, to build up *models* for natural systems
- In the sciences of artificial, noumena are to be created
  - definition is the foundation for systems, and gives *structure* to *artificial worlds*
  - there, Occam's Razor and the Lex Parsimonia may apply to definition instead of theory and explanation

Lessons Learned: Definition by Genus and Differentia

Some rules of thumb

genus A definition should clearly delimit the domain of discourse

differentia A definition should allow what is in and what is out to be clearly determined

rules A definition should follow the rules for definition by genus and differentia

- essentiality, no circularity, neither wide nor narrow, no obscurity, no unneeded negativity

Lessons Learned: Occam's Razor & Lex Parsimoniae

Other rules of thumb

minimal assumptions A definition of an entity should make as few assumptions as possible

minimal complication Given two equally valid definitions for an entity, one should embrace the less complicated formulation

lex parsimoniae Definitions should not be multiplied beyond necessity

- definitory features should not be multiplied beyond necessity

Autonomy as the Foundation of the Definition of Agent

Lex Parsimoniae: Autonomy

- Autonomy as the only fundamental and definitory feature of agents
- Let us see whether other typical agent features follow / descend from this somehow

Computational Autonomy

- Agents are autonomous as they *encapsulate* (the thread of) *control*
- Control does not pass through agent boundaries
  - only data (knowledge, information) crosses agent boundaries
- Agents have no interface, cannot be controlled, nor can they be invoked
- Looking at agents, MAS can be conceived as an aggregation of multiple distinct *loci* of control interacting with each other by exchanging information

(Autonomous) Agents (Pro-)Act

Action as the essence of agency

- The etimology of the word *agent* is from the Latin agens
- So, agent means "the one who acts"
- Any coherent notion of agency should naturally come equipped with a model for agent actions

Autonomous agents are pro-active

- Agents are literally active
- Autonomous agents encapsulate control, and the rule to govern it

$\longrightarrow$ Autonomous agents are pro-active by definition

  - where pro-activity means "making something happen", rather than waiting for something to happen

Agents are Situated

The model of action depends on the context

- Any "ground" model of action is strictly coupled with the context where the action takes place
- An agent comes with its own model of action
- Any agent is then strictly coupled with the environment where it lives and (inter)acts
- Agents are in this sense are *situated*

Are Agents Reactive?

Situatedness and reactivity come hand in hand

- Any model of action is strictly coupled with the context where the action takes place
- Any action model requires an adequate *representation* of the world
- Any *effective* representation of the world requires a *suitable* balance between environment *perception* and representation

→ Any effective action model requires a suitable balance between environment perception and representation

  - however, any non-trivial action model requires some form of perception of the environment—so as to check action pre-conditions, or to verify the effects of actions on the environment

- Agents in this sense are supposedly *reactive* to change

Are Autonomous Agents Reactive?

Reactivity as a (deliberate) reduction of proactivity

- An autonomous agent could be built / choose to merely react to external events
- It may just wait for something to happen, either as a permanent attitude, or as a temporary opportunistic choice
- In this sense, autonomous agents may also be reactive

Reactivity to change

- Reactivity to (environment) change is a different notion
- This mainly comes from early AI failures, and from robotics
- It stems from agency, rather than from autonomy—as discussed in the previous slide
- However, this issue will be even clearer when facing the issue of artifacts and environment design

(Autonomous) Agents Change the World

Action, change & environment

- Whatever the model, any model for action brings along the notion of *change*
  - an agent acts to change something around in the MAS
- Two admissible targets for change by agent action

  agent an agent could act to change the state of another agent
    - since agents are autonomous, and only data flow among them, the only way another agent can change their state is by providing them with some information
    - change to other agents essentially involves *communication actions*

  environment an agent could act to change the state of the environment
    - change to the environment requires *pragmatical actions*
    - which could be either physical or virtual depending on the nature of the environment

Autonomous Agents are Social

From autonomy to society

- From a philosophical viewpoint, autonomy only makes sense when an individual is immersed in a society
  - autonomy does not make sense for an individual in isolation
  - no individual alone could be properly said to be autonomous
- This also straightforwardly explain why any program in any sequential programming language is not an autonomous agent *per se* [Graesser, 1996, Odell, 2002]

Autonomous agents live in a MAS

- Single-agent systems do not exist in principle
- Autonomous agents live and interact within agent societies & MAS
- Roughly speaking, MAS are the only "legitimate containers" of autonomous agents

Autonomous Agents are Interactive

Interactivity is not a definitory feature

- Since agents are subsystems of a MAS, they interact within the global system
  - by essence of systems in general, rather than of MAS
- Since agents are autonomous, only data (knowledge, information) crosses agent boundaries
- Information & knowledge is exchanged between agents
  - leading to more complex patterns than message passing between objects

Autonomous Agents Do not Need a Goal or a Task

Agents govern MAS computation

- By encapsulating control, agents are the main forces governing and pushing computation, and determining behaviour in a MAS
- Along with control, agent should then encapsulate the *criterion* for regulating the thread(s) of control

Autonomy as self-regulation

- The term "autonomy", at its very roots, means self-government, self-regulation, self-determination
  - "internal unit invocation" [Odell, 2002]
- This does *not* imply in any way that agents *needs* to have a goal, or a task, to be such—to be an agent, then
- However, this *does* imply that autonomy captures the cases of goal-oriented and task-oriented agents
  - where goals and tasks play the role of the criteria for governing control

Goal-/Task-Orientedness is not a Definitory Feature for Agents

Example: finite-state automaton with encapsulated control

- An agent might be a finite-state automaton
- Encapsulating control as an independent thread
- Equipped with state transition rules
- The criteria for the govern of control would there be embodied in terms of (finite) states and state transition rules

Goal-orientedness and task-orientedness are just possible features for agents

- They are not definitory features anyway

Are Autonomous Agents Intelligent?

Intelligence helps autonomy

- Autonomous agents have to self-determine, self-govern, ...
- Intelligence makes it easy for an agent to govern itself
- *Intelligent autonomous agents* clearly make sense
  - intelligence, however, is *not* required for an agent to be autonomous

Are Autonomous Agents Mobile?

Mobility is an extreme form of autonomy

- Autonomous agents encapsulate control
- At the end of the story, control might be independent of the environment where an agent lives—say, the virtual machine on which it runs
- *Mobile autonomous agents* clearly make sense
  - mobility, however, is *not* required for an agent to be autonomous

Do Autonomous Agents Learn?

Learning may improve agent autonomy

- By learning, autonomous agents may acquire new skills, improve their practical reasoning, etc.
- In short, an autonomous agent could learn how to make a better use out of its autonomy
- *Learning autonomous agents* clearly make sense
  - learning, however, is *not* required for an agent to be autonomous

Agents in the A&A Meta-model

Definition (A&A Agent)

An A&A agent is an *autonomous computational entity*

genus
: agents are computational entities

differentia
: agents are autonomous, in that they encapsulate control along with a criterion to govern it

A&A agents are *autonomous*

- From autonomy, many other features stem
  - autonomous agents *are* interactive, social, proactive, and situated;
  - they *might* have goals or tasks, or be reactive, intelligent, mobile
  - they live within MAS, and *interact* with other agents through *communication actions*, and with the environment with *pragmatical actions*

Artifacts in the A&A Meta-model

Definition (A&A Artifact)

An A&A artifact is a *computational entity* aimed at the *use* by A&A agents

genus: artifacts are computational entities

differentia: artifacts are aimed to be used by agents

Artifacts are *to be used* by agents

- From use, many other features stem
  - which are either essential or desirable, but need not to be used as definitory ones

Artifacts Have a Function

Artifacts are designed for use

- Being *aimed at* the agent's *use*, artifacts are *designed* to serve some purpose
  - and built as such
- When designed, they are then associated by design to their *function*
- Artifact function does not necessarily determine the actual use of the artifact by an agent
  - however, it incorporates the *aim* of the artifact designer, envisioning the artifact as potentially serving agent's purposes

Artifacts are transparent & predictable

transparency In order to be used by agents, artifact function should be available to / understood by agents

predictability In order to promote agent's use, artifact behaviour should be predictable

Artifacts Are Not Autonomous

Artifacts are designed to serve

- Artifacts are designed to *serve* some agent's purpose
  - not to follow their own path of action
- An artifact has an embodied function, made repeatedly and predictably available to agents
- An artifact is a tool in the "hands" of agents
  - it does not need to be self-governed, it just has to be "governed" by agents when they use it

Artifacts Are (Computationally) Reactive

Artifacts are *reactive* in terms of control

- Artifacts behave in response to agent use
  - the behaviour of an artifact just needs to emerge when it is used by an agent
- In terms of control, an artifact just needs to be *reactive*
  - or, to behave as it were
- What about reaction to change?
  - should artifacts be reactive to environment change?

Artifacts Have Operations and Interfaces

Agents use artifact operations

- In order to be used, artifacts should make *operations* available to agents
- Operations change an artifact's state, make it behave and produce the desired effects on the environment
- Either explicitly or implicitly, an artifact exhibits its *interface* to agents, as the collection of the operations made available

Artifacts are Situated

Artifacts & Agent Actions

- Being used, artifacts are the primary target / means of agent's action
  - action is what makes agents strictly coupled with the environment
- Artifact's function is expressed in terms of change to the environment
  - what the artifact actually *does* when used
- Artifact's model, structure & behaviour are *expressed* in terms of agent's actions and *environment*
  - artifacts are *situated*

Artifacts are reactive to change

- Along the same line used for agents, artifacts are then supposedly *reactive to change*
  - since they are structurally reactive in computational terms, this comes for free—unlike (proactive) agents

Artifacts Are Not Agents

Agents vs. artifacts

- Agents are autonomous, artifacts are not
- Agents encapsulate control, artifacts do not
- Agents are proactive, artifacts are not
- Agents are opaque, artifacts are transparent
- Artifacts are predictable, agents are not
- Agents may have a goal / task, artifacts do not
- Artifacts have a function, agents have not
- Agents use artifacts, but cannot use agents
- Agents speak with agents, but cannot speak with artifacts
- Agents are designed to govern, artifacts are designed to serve

Artifacts in the A&A Meta-model

Definition (A&A Artifact)

An A&A artifact is a *computational entity* aimed at the *use* by A&A agents

genus
: artifacts are computational entities

differentia
: artifacts are aimed to be used by agents

Artifacts are *to be used* by agents

- From use, many other features stem
  - artifacts have a function, are computationally reactive, are situated and reactive to change, are not autonomous, are transparent and predictable, have operations and interface for agent's use
  - artifacts are not agents

Artifacts & Environment

Artifacts as mediators

- Artifacts mediate between agents and the environment
- Artifacts embody the portion of the environment that can be designed and controlled to support MAS activities

Artifacts as representatives of MAS environment

- As an observable & controllable part of the environment, artifacts can be monitored along with the development of MAS activities
  - to evaluate overall MAS performance
  - to keep track of MAS history
  - to influence MAS behaviour and evolution

Artifacts for environment design

- Artifacts are the essential tools
  - for modelling MAS environment
  - to shape MAS environment so as to make it favourable to the development of MAS social activities

Artifacts as Enablers and Constrainers of MAS Activities

- As mediating tools, artifacts have both an *enabling* and a *constraining* function

  enablers: artifacts expand out agent's ability to manipulate and transform different objects

  constrainers: the environment is perceived and manipulated by agents through the artifact not 'as such' but within the limitations set by the artifact itself

- A simple example: an agent-oriented printer driver

  enabler: enables agents to use a printer, along with a number of its options

  constrainer: limits in general agent interaction with the printer to some well-defined interaction patterns

Desirable Features of A&A Artifacts

How do we like artifacts?

- Artifacts could exhibit a number of relevant features, which would in principle enhance MAS engineers / agents ability to use them for their own purposes [Omicini et al., 2006]
  - *inspectability*
  - *controllability*
  - *malleability / forgeability*
  - *predictability*
  - *formalisability*
  - *linkability*
  - *distribution*

A&A Artifacts: Inspectability

- The state of an artifact, its content (whatever this means in a specific artifact), its operations, interface and function might be all or partially available to agents through *inspectability*
- Whereas in closed MASs this information could be hard-coded in the agent—the artifact engineer develops the agents as well—, in open MASs third-party agents should be able to dynamically join a society and get aware at run-time of the necessary information about the available artifacts
- Also, artifacts are often in charge of critical MAS behaviour [Omicini et al., 2004a]: being able to inspect a part or the whole of an artifact features and state is likely to be a fundamental capability in order to understand and govern the dynamics and behaviour of a MAS

A&A Artifacts: Controllability

- *Controllability* is an obvious extension of the inspectability property
- The operational behaviour of an artifact should then not be merely inspectable, but also *controllable* so as to allow MAS engineers (or even intelligent agents) to monitor its proper functioning
  - it should be possible to stop and restart an artifact working cycle, to trace its inner activity, and to observe and control a step-by-step execution
- In principle, this would largely improve the ability of monitoring, analysing and debugging the operational behaviour of an artifact at execution time, and of the associated MAS social activities as well

A&A Artifacts: Malleability

- Also related to inspectability, *malleability* (also called *forgeability*) is a key-feature in dynamic MAS scenarios, when the behaviour of artifacts could require to be modified dynamically in order to adapt to the changing needs or mutable external conditions of a MAS
- Malleability, as the ability to change the artifact behaviour at execution time, is seemingly a crucial aspect in on-line engineering for MASs, and also a perspective key-issue for self-organising MASs

A&A Artifacts: Predictability

- Differently from agents—which as autonomous entities have the freedom of behaving erratically, e.g. neglecting messages—, artifact operations, interface and function description can be used as the stable basis for a contract between an artifact and an agent
- In particular, the description of the artifact function could provide precise details of the outcomes of exploiting the artifact, while description of the artifact operations, interface and behaviour should make the behaviour of an artifact *predictable* for an agent

A&A Artifacts: Formalisability

- The predictability feature can be easily related with *formalisability*
- Due to the precise characterisation that can be given to an artifact behaviour, until reaching e.g. a full operational semantics model—for instance, as developed for coordination artifacts in [Omicini et al., 2004b]—it might be feasible to automatically verify the properties and behaviour of the services provided by artifacts, for this is intrinsically easier than services provided by autonomous agents

A&A Artifacts: Linkability

- Artifacts can be used encapsulate and model reusable services in a MAS
- To scale up with complexity of an environment, it might be interesting to compose artifacts, e.g. to build a service incrementally on top of another, by making a new artifact realising its service by interacting with an existing artifact
- To this end, artifacts should be able to invoke the operation of another artifact: the reply to that invocation will be transmitted by the receiver through the invocation of another operation upon the caller

A&A Artifacts: Distribution

- Differently from an agent, which is typically seen as a point-like abstraction conceptually located to a single node of the newtwork, artifacts can also be distributed
- In particular, a single artifact can in principle be used to model a distributed service, accessible from more nodes of the net
- Using linkability, a distributed artifact can then be conceived and implemented as a composition of linked, possibly non-distributed artifacts—or viceversa, a number of linked artifacts, scattered through a number of different physical locations could be altogether seen as a single distributed artifact
- Altogether, distribution and linkability promote the *layering* of artifact engineering—as sketched in [Molesini et al., 2006]

Levels of Use of Artifacts

Co-ordination: both intelligent and non-intelligent agents could coordinate

Any agent (either intelligent or not) could simply exploit artifacts to achieve its own goals by simply taking artifacts as they are, and use them

Co-operation: intelligent agents could change artifacts to change MAS

Intelligent agents could possibly reason about the nature of the artifacts as well as on the level of achievement of their goals, and take the chance to change or adapt the artifacts, or even to create new ones whenever useful and possible as the result of either an individual or a social activity

Co-operation: MAS engineers could embody social intelligence in artifacts

In the same way, MAS engineers can use artifacts to embody the “social intelligence” that actually characterises the systemic/synergistic (as opposed to compositional) vision of MAS , but also to observe, control, and possibly change MAS social behaviour [Ciancarini et al., 2000]

Agents & Artifacts Interacting

Aspects of agent-artifact relationship

use An agent can use an artifact, according to its use goal, associating it with a destination

- **aware use** because the agent is aware of the artifact's function
- **unaware use** because the artifact's use is encoded in the agent by the programmer / designer

selection An agent can select an artifact for future use, according to its use-value goal, reasoning about its possible future destinations and use goals

construction / manipulation An agent can modify an artifact to adapt its function to some required use-value goals and to its possible future destinations

- or, an agent can create *ex-novo* a new artifact with an agent-designed function according to some required use-value goals and to its possible future destinations

A Model of A&A Artifacts

Rational exploitation of artifacts by intelligent agents

- In order to allow for its rational exploitation by intelligent agents, an A&A artifact possibly exposes
  - a *usage interface*
  - *operating instructions*
  - a *function description*

A&A Artifacts: Usage Interface

Agents, artifacts & operations

- One of the core differences between artifacts and agents is the concept of *operation*
- An operation is the means by which an artifact provides agents with a service or function
- An agent executes an action over an artifact by invoking an artifact operation
- Execution possibly terminates with an *operation completion*, typically representing the outcome of the invocation, which the agent comes to be aware of in terms of *perception*

usage interface The set of operations provided by an artifact defines what is called its *usage interface*

- which (intentionally) resembles interfaces of services, components or objects—in the object-oriented acceptation of the term

A&A Artifacts: Operating Instructions

Artifact's manuals for intelligent agents

- Operations cannot be invoked in any order
- Artifact's state & behaviour, along with the effects of agent's actions on the environment via the artifact, depend on the execution order of operations

operating instructions *Operating instructions* are a description of the procedure an agent has to follow to meaningfully interact with an artifact over time

- which should of course be coupled with usage interface

- Operating instructions are a description of the possible *usage protocols*, i.e. sequences of operations that can be invoked on the artifact, in order to exploit its function
- Besides a syntactic information, they can also embed some sort of semantic information for rational agents
  - rational agents can use such information for their practical reasoning
- Artifacts are conceptually similar to devices used by humans
  - operation instructions play for agents a role similar to a manual for a human—which a human reads to know how to use the device on a step-by-step basis, and depending on the expected outcomes he/she needs to achieve

A&A Artifacts: Function Description

Agents, artifacts & function

- Agents should be provided with a description of the functionality provided by the artifact
  - which agents essentially use for artifact selection

function description Artifacts could then be equipped with a *function description* (or, a *service description*), (formally) describing the function / service that the artifact is designed to provide agents with

- differently from operating instructions, which describes *how* to exploit an artifact, function description describes *what* to obtain from an artifact

An example

When modelling a sensor wrapper as an artifact, we may easily think of the operations for sensor activation and inspection as described via usage interface and operations instructions, while the information about the sensory function itself being conveyed through function description of the sensor wrapper

MAS in the A&A Meta-model

Definition (A&A MAS)

An A&A MAS is a *computational systems* made of agents and artifacts

genus MAS is computational system

differentia its basic components are agents and artifacts

A constructive definition

- Based on the previous definitions
- Also based on on the (primitive) notion of system as well

A&A MAS are Situated

MAS & situatedness

- MAS are made of agents & artifacts
- Both agents & artifacts are situated computational entities
- As an obvious consequence, MAS are *situated computational systems*

MAS & environment

- A MAS is always immersed within an environment
- A MAS cannot be conceived / modelled / designed in a separate way with respect to its environment

A&A MAS have a Behaviour

MAS & activity

- MAS are made of agents & artifacts
- Agents are pro-active, artifacts are reactive
- Agents are autonomous entities, artifacts have functions

$\longrightarrow$ In the overall, a MAS has a behaviour that results from the interaction of autonomous, self-governing entities (agents) and reactive, functional entities (artifacts)

MAS Interaction in the A&A Meta-model

Admissible interactions *within* a MAS

- MAS are made of agents & artifacts
- Two fundamental entities give raise to four different sorts of admissible interactions

communication agents *speak* with agents
operation agents *use* artifacts
composition artifacts *link* with artifacts
presentation artifacts *manifest* to agents

MAS interactions with the environment

- Defining a system is to define a boundary—the same holds for a MAS, of course
- Interactions occur within and without the boundaries
  - MAS interaction with the environment
- Depending on the desired level of abstraction, we may attribute environment interactions to either individual agents & artifacts, or to the MAS as a whole

Delimiting a MAS

MAS boundaries

- Our definition allows us to understand whether a computational system is a MAS
- It mostly define the class of the MAS in the A&A meta-model

What is an open system?

- How can we determine / recognise the boundaries of an open MAS?
- On the engineering side, how can we design an open MAS?
  - what should we actually design when designing a MAS?
  - what should anyway account for / account not?

Essence of a single MAS

MAS characteristic

- To define one single MAS, we need a characterising criterion
- The very notion of system means there is a coherent way to interpret the overall set of components as a whole, and to determine whether a given component belongs to a given MAS
- Characterising a single MAS then means firstly to define a criterion according to which an agent / an artifact could be said either to belong or not to a given MAS
  - hopefully in a univocal way
  - possibly dynamically depending on a number of parameters, like time, state of components, state of MAS, state of the environment, ...

Bibliography I

Ciancarini, P., Omicini, A., and Zambonelli, F. (2000).
Multiagent system engineering: The coordination viewpoint.
In Jennings, N. R. and Lespérance, Y., editors, *Intelligent Agents VI. Agent Theories, Architectures, and Languages*, volume 1757 of *LNAI*, pages 250–259. Springer-Verlag.
6th International Workshop (ATAL'99), Orlando, FL, USA, 15–17 July 1999. Proceedings.

Graesser, S. F. A. (1996).
Is it an agent, or just a program?: A taxonomy for autonomous agents.
In Jörg P. Müller, Michael J. Wooldridge, N. R. J., editor, *Intelligent Agents III Agent Theories, Architectures, and Languages: ECAI'96 Workshop (ATAL) Budapest, Hungary, August 12–13, 1996 Proceedings*, volume 1193 of *Lecture Notes In Computer Science*, pages 21 – 35. Springer.

Molesini, A., Omicini, A., Ricci, A., and Denti, E. (2006).
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 International Workshop (AOSE 2005), Utrecht, The Netherlands, 25–26 July 2005. Revised and Invited Papers.

Odell, J. (2002).
Objects and agents compared.
Journal of Object Technologies, 1(1):41–53.

Bibliography II

Omicini, A., Ossowski, S., and Ricci, A. (2004a).
Coordination infrastructures in the engineering of multiagent systems.
In Bergenti, F., Gleizes, M.-P., and Zambonelli, F., editors, *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, volume 11 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, chapter 14, pages 273–296. Kluwer Academic Publishers.

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.

Omicini, A., Ricci, A., Viroli, M., Castelfranchi, C., and Tummolini, L. (2004b).
Coordination artifacts: Environment-based coordination for intelligent agents.
In Jennings, N. R., Sierra, C., Sonenberg, L., and Tambe, M., editors, *3rd international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, volume 1, pages 286–293, New York, USA. ACM.

Agents and Artifacts: The A&A Meta-model for Multiagent Systems

Multiagent Systems LS
Sistemi Multiagente LS

Andrea Omicini
andrea.omicini@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2007/2008