Computational Agents & Artefacts

Dottorato di Ricerca in Informatica Giuridica e Diritto dell'Informatica

Andrea Omicini
andrea.omicini@unibo.it

CIRSFID
Alma Mater Studiorum—Università di Bologna

Academic Year 2009/2010

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

The Change is Widespread

- [Zambonelli and Parunak, 2003]
- Today software systems are essentially different from "traditional" ones
- The difference is widespread, and not limited to some application scenarios

Computer science & software engineering are going to change

- dramatically
- complexity is too *huge* for traditional CS & SE abstractions
  - like object-oriented technologies, or component-based methodologies

The Next Crisis of Software

The Scenario of the Crisis

Computing systems

- will be anywere
  - will be embedded in every environment item/ object
- always connected
  - wireless technologies will make interconnection pervasive
- always active
  - to perform tasks on our behalf

Impact on Software Engineering

Which impact on the design & development of software systems?

- Quantitative
  - in terms of computational units, software components, number of interconnections, people involved, time required, ...
  - current processes, methods and technologies do not scale
- Qualitative
  - new software systems are *different* in kind
  - new features never experimented before

Novel Features of Complex Software Systems

- Situatedness
  - computations occur within an environment
  - computations and environment mutually affect each other, and cannot be understood separately
- Openness
  - systems are permeable and subject to change in size and structure
- Locality in control
  - components of a system are autonomous and proactive *loci* of control
- Locality in interaction
  - components of a system interact based on some notion of spatio-temporal compresence on a *local* basis

Examples

Fields like

- distributed artificial intelligence
- manufacturing and environmental control systems
- mobile computing
- pervasive / ubiquitous computing
- Internet computing
- peer-to-peer (P2P) systems

have already registered the news, and are trying to account for this in technologies and methodologies

Situatedness—Examples

Control systems for physical domains

- manufacturing, traffic control, home care, health care systems
- explicitly aim at managing / capturing data from the environment through event-driven models / event-handling policies

Sensor networks, robot networks

- are typically meant to sense, explore, monitor and control partially known / unknown environments

Situatedness I

Environment as a first-class entity

- the notion of *environment* is explicit
- components / computations interact with, and are affected by the environment
- interaction with the environment is often explicit, too

Is this new?

- every computation always occurred in some context
- however, the environment is *masked* behind some "wrapping" abstractions
- environment is not a *primary* abstraction

Situatedness II

Does masking / wrapping work?

- wrapping abstractions are often too simple to capture complexity of the environment
- when you need to sense / control the environment, masking it is not always a good choice
- environment dynamics is typically independent of system dynamics
  - the environment is often unpredictable and non-formalisable [Wegner, 1997]

Situatedness III

Trend in CS and SE

- drawing a line around the system
- explicitly representing
  - what is inside in terms of component's behaviour and interaction
  - what is outside in terms of environment, and system interaction with the environment
- predictability of components vs. unpredictability of the environment
  - this dichotomy is a key issue in the engineering of complex software systems

Openness—Examples

Critical control systems

- unstoppable systems, run forever
- they need to be adapted / updated anyway, in terms of either computational or physical components
- openness to change, and automatic reorganisation are essential features

Systems based on mobile devices

- the dynamics of mobile devices is out of the system / engineer's control
- system should work without assumptions on presence / activity of mobile devices
- the same holds for Internet-based / P2P systems

Openness

Permeable boundaries

- drawing lines around "systems" does not make them isolated
- boundaries are often just conventional, thus allow for mutual interaction and side-effects

The dynamics of change

- systems may change in structure, cardinality, organisation, ...
- technologies, methodologies, but above all abstractions should account for modelling (possibly governing) the dynamics of change

Openness—Further Issues

Where is the system?

- where do components belong?
- are system boundaries for real?

Mummy, where am I?

- how should components become aware of their environment?
- when they enter a system / are brought to existence?

How do we control open systems?

- . . . where components come and go?
- . . . where they can interact at their will?

Local Control—Examples

Cellular phone network

- each cell with its own activity / autonomous control flow
- autonomous (inter)acting in a world-wide network

World Wide Web

- each server with its own (reactive) independent control flow
- each browser client with its own (proactive) independent control flow

Local Control

Flow of Control

- key notion in traditional systems
- key notion in Computer Science
- multiple flows of control in concurrent / parallel computing
- however, not an immediate notion in complex software systems
  - a more general / powerful notion is required

Autonomy

- is the key notion here
- subsuming control flow / motivating multiple, independent flows of control
- at a higher level of abstraction

Local Control—Issues of Autonomy

- in an open world, autonomy of execution makes it easy for components to move across systems & environments
- autonomy of components more effectively matches dynamics of environment
- autonomy of execution is a suitable model for multiple independent computational entities
- SE principles of locality and encapsulation cope well with delegation of control to autonomous components

Local Interactions—Examples

Control systems for physical domains

- each control component is delegated a portion of the environment to control
- interactions are typically limited to the neighboring portions of the environment
- strict coordination with neighboring components is typically enforced

Mobile applications

- local interaction of mobile devices is the basis for "context-awareness"
- interactions are mostly with the surrounding environment
- interoperation with neighboring devices is typically enabled

Local Interactions

Local interactions in a global world

- autonomous components interact with the environment where they are located
  - interaction is limited in extension by either physical laws or logical constraints
- autonomous components interact openly with other systems
  - motion to and local interaction within the new system is the cheapest and most suitable model
- situatedness of autonomous components calls for context-awareness
  - a notion of locality is required to make context manageable

Summing Up

Complex software systems, then

- made of autonomous components
- locally interacting with each other
- immersed in an environment—both components and the system as a whole
- system / component boundaries are blurred—they are conceptual tools until they work

Change is ongoing

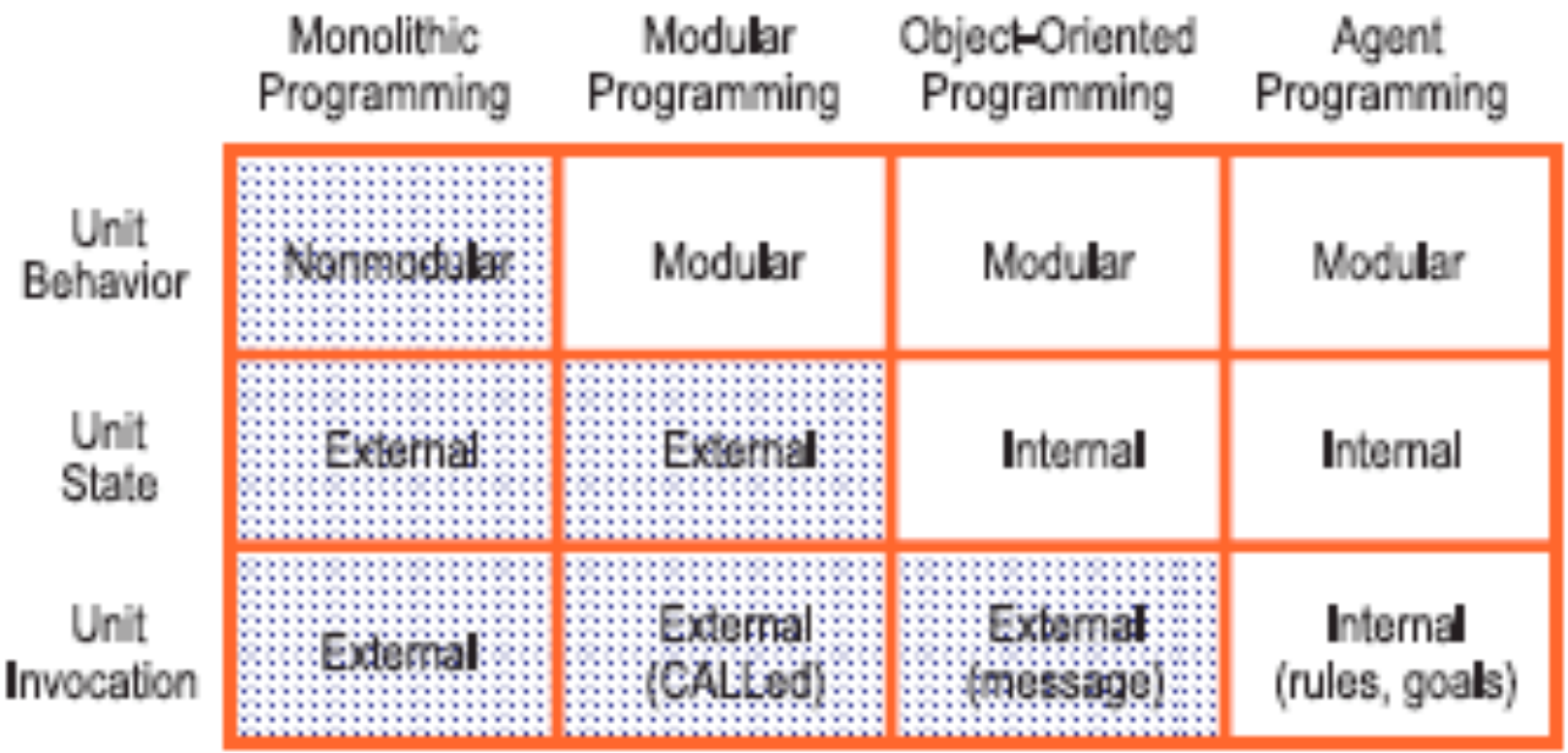
- Computer Science is changing
- Software Engineering is changing
- a (sort of) paradigm shift is occurring—a *revolution*, maybe? [Kuhn, 1996]

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Evolution of Programming Languages: The Picture

- [Odell, 2002]



Evolution of Programming Languages: Dimensions

Historical evolution

- Monolithic programming
- Modular programming
- Object-oriented programming
- Agent programming

Degree of modularity & encapsulation

- Unit behaviour
- Unit state
- Unit invocation

Monolithic Programming

- The basic unit of software is the whole program
- Programmer has full control
- Program's state is responsibility of the programmer
- Program invocation determined by system's operator
- Behaviour could not be invoked as a reusable unit under different circumstances
  - modularity does not apply to unit behaviour

Evolution of Programming Languages: The Picture

Monolithic Programming

Encapsulation? There is no encapsulation of anything, in the very end

	Monolithic Programming	Modular Programming	Object-Oriented Programming	Agent Programming
Unit Behavior	Nonmodular	Modular	Modular	Modular
Unit State	External	External	Internal	Internal
Unit Invocation	External	External (CALLed)	External (message)	Internal (rules, goals)

The Prime Motor of Evolution

Motivations

- Larger memory spaces and faster processor speed allowed program to became more complex

Results

- Some degree of organisation in the code was required to deal with the increased complexity

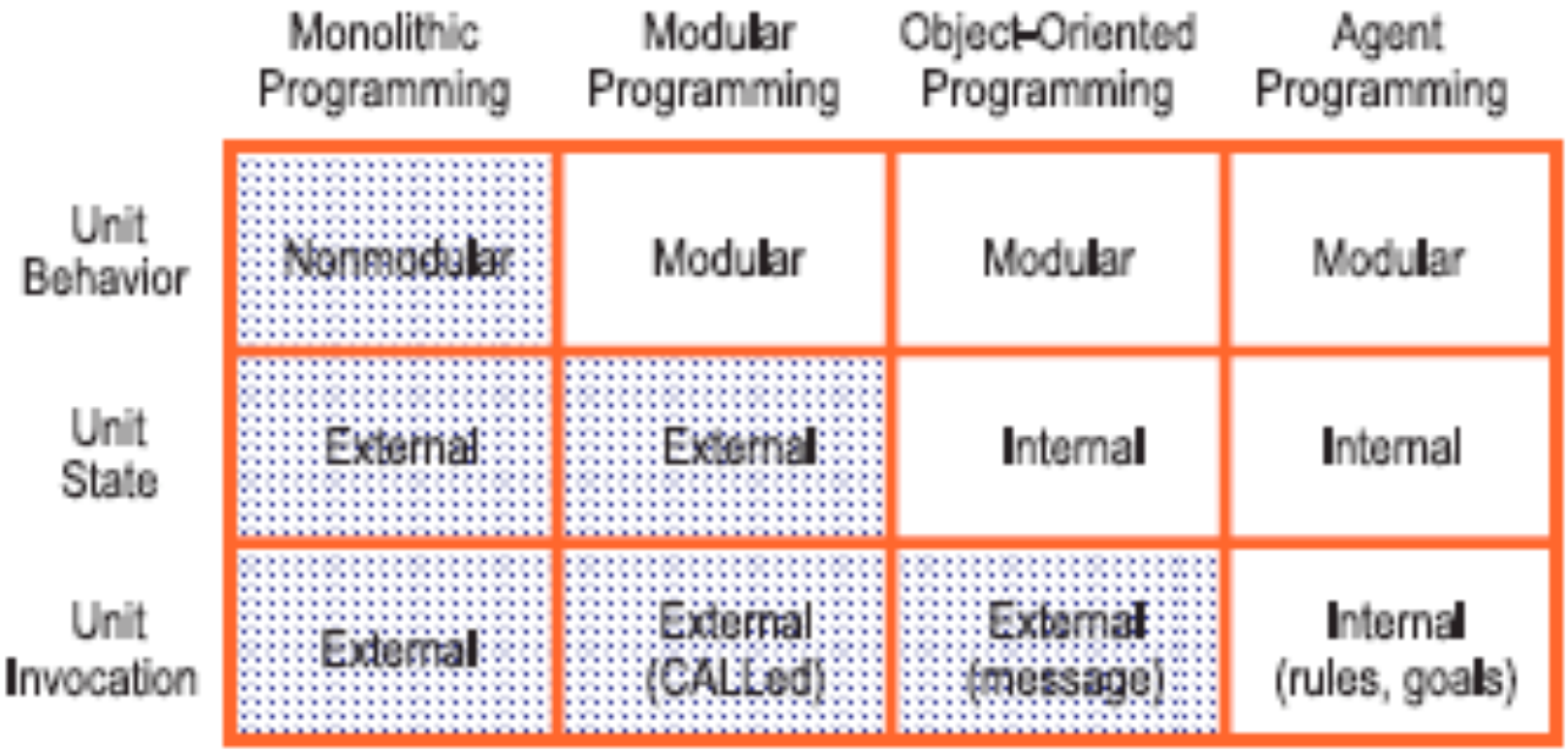
Modular Programming

- The basic unit of software are structured loops / subroutines / procedures / ...
  - this is the era of procedures as the primary unit of decomposition
- Small units of code could actually be reused under a variety of situations
  - modularity applies to subroutine's code
- Program's state is determined by externally supplied parameters
- Program invocation determined by CALL statements and the likes

Evolution of Programming Languages: The Picture

Modular Programming

Encapsulation? Encapsulation applies to *unit behaviour* only



Object-Oriented Programming

- The basic unit of software are objects & classes
- Structured units of code could actually be reused under a variety of situations
- Objects have local control over variables manipulated by their own methods
  - variable state is persistent through subsequent invocations
  - object's state is encapsulated
- Object are passive—methods are invoked by external entities
  - modularity does not apply to unit invocation
  - object's control is not encapsulated

Evolution of Programming Languages: The Picture

Object-Oriented Programming

Encapsulation? Encapsulation applies to unit *behaviour* & *state*

	Monolithic Programming	Modular Programming	Object-Oriented Programming	Agent Programming
Unit Behavior	Nonmodular	Modular	Modular	Modular
Unit State	External	External	Internal	Internal
Unit Invocation	External	External (CALLed)	External (message)	Internal (rules, goals)

Agent-Oriented Programming

- The basic unit of software are agents
  - encapsulating everything, in principle
    - by simply following the pattern of the evolution
  - whatever an agent is
    - we do not need to define them now, just to understand their desired features
- Agents could in principle be reused under a variety of situations
- Agents have control over their own state
- Agents are active
  - they cannot be invoked
  - agent's control is encapsulated

Evolution of Programming Languages: The Picture

Agent-Oriented Programming

Encapsulation? Encapsulation applies to unit *behaviour*, *state* & *invocation*

	Monolithic Programming	Modular Programming	Object-Oriented Programming	Agent Programming
Unit Behavior	Nonmodular	Modular	Modular	Modular
Unit State	External	External	Internal	Internal
Unit Invocation	External	External (CALLed)	External (message)	Internal (rules, goals)

Features of Agents

Before we define agents...

- ...agents are *autonomous* entities
  - encapsulating their thread of control
  - they can say "Go!"
- ...agents cannot be invoked
  - they can say "No!"
  - they do not have an interface, nor do they have methods
- ...agents need to encapsulate a criterion for their activity
  - to self-govern their own thread of control

Dimensions of Agent Autonomy

Dynamic autonomy

- Agents are *dynamic* since they can exercise some degree of activity
  - they can say "Go!"
- From passive through reactive to active

Unpredictable / non-deterministic autonomy

- Agents are *unpredictable* since they can exercise some degree of deliberation
  - they can say "Go!", they can say "No!"
  - and also because they are "opaque"—may be unpredictable to external observation, not necessarily to design
- From predictable through partially predictable to unpredictable

Objects vs. Agents: Interaction & Control

Message passing in object-oriented programming

- Data flow along with control
  - data flow cannot be designed as separate from control flow
- A too-rigid constraint for complex distributed systems...

Message passing in agent-oriented programming

- Data flow through agents, control does not
  - data flow can be designed independently of control
- Complex distributed systems can be designed by designing information flow

Agents Communication

Agents communicate

- Interaction between agents is a matter of exchanging information
  - toward Agent Communication Languages (ACL)
- Agents can be involved in *conversations*
  - they can be involved in associations lasting longer than the single communication act
  - differently from objects, where one message just refer to one method

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Convergence Towards The Agent

Many areas contribute their own notion of agent

- Artificial Intelligence (AI)
- Distributed Artificial Intelligence (DAI)
- Parallel & Distributed Systems (P&D)
- Mobile Computing
- Programming Languages and Paradigms (PL)
- Software Engineering (SE)
- Robotics

On the Notion of Intelligence in AI

Reproducing intelligence

- AI is first of all concerned with reproducing *intelligent processes* and *behaviours*, where
  - intelligent processes roughly denote *internal* intelligence—like understanding, reasoning, representing knowledge, ...
  - intelligent behaviours roughly represent *external*, observable intelligence—like sensing, acting, communicating, ...

Symbolic intelligence

- Classic AI promoted the so-called *symbolic* acceptation of (artificial) intelligence
  - based on *mental representation* of the external environment
  - where the environment is typically oversimplified
  - and the agent is the only source of disruption

On the Notion of Agent in AI

Encapsulating intelligence

- Agents in AI have from the very beginning worked as the units encapsulating *intelligence*
  - *individual* intelligence
  - within the symbolic interpretation of intelligence

Cognitive agents

- AI agents are essentially cognitive agents
  - they are *first* cognitive entities
  - *then* active entities
  - in spite of their very name, coming from Latin *agens* [*agere*]—the one who acts

AI & Agents—A Note

Reversing perspective [Omicini and Poggi, 2006]

- Today, results from AI and MAS research are no longer so easily distinguishable
- Agents and MAS have become the introductory metaphors to most of the AI results
  - as exemplified by one of the most commonly used AI textbooks [Russell and Norvig, 2002]
- Classic AI results on planning, practical reasoning, knowledge representation, machine learning, and the like, have become the most obvious and fruitful starting points for MAS research and technologies
- It is quite rare nowadays that new findings or lines of research in AI might ignore the agent abstractions at all
- Altogether, rather than a mere subfield of AI, agents and MAS could be seen as promoting a new paradigm, providing a new and original perspective about computational intelligence and intelligent systems

On the Notion of Agent in DAI [Wooldridge, 2002]

Overcoming the individual dimension

- no more a single unit encapsulating individual intelligence
- and acting *alone* within an oversimplified environment

Social acceptation of agency

- agents are individuals within a society of agents
  - agents are components of a *multiagent system* (MAS)
- agents are distributed within a *distributed environment*

Agent Features in DAI [O'Hare and Jennings, 1996]

A DAI agent...

- ...has an explicit representation of the world
- ...is situated within its environment
- ...solves a problem that requires intelligence
- ...deliberates / plans its course of actions
- ...is flexible
- ...is adaptable
- ...learns

A DAI Agent Represents the World: What?

What should be represented?

- What is relevant? What is not relevant?
  - More precisely, which knowledge about the environment is relevant for an agent to effectively plan and act?
  - So, which portion of the environment should the agent explicitly represent somehow in order to have the chance to behave intelligently?

Representation is partial

- Necessarily, an agent has a *partial* representation of the world
- Its representation includes in general both the current *state* of the environment, and the *laws* regulating its dynamics

A DAI Agent Represents the World: How?

The issue of Knowledge Representation (KR)

- How should an agent represent knowledge about the world?
- Representation is not neutral with respect to the agent's model and behaviour
  - and to the engineer's possibilities as well
- Choosing the right KR language / formalism
  - according to the agent's (conceptual & computational) model
  - multisets of tuples, logic theories, description logics, ...?

A DAI Agent Represents the World: Consistency I

Perception vs. representation

- Environment changes, either by agent actions, or by its own dynamics
- Even supposing that an agent has the potential to observe all the relevant changes in the environment, it can not spend all of its activity monitoring the environment and updating its internal representation of the world
- So, in general, how could consistency of internal representation be maintained? And to what extent?
  - in other terms, how and to what extent can an agent be ensured that its knowledge about the environment is at any time consistent with its actual state

A DAI Agent Represents the World: Consistency II

Reactivity vs. proactivity

- An agent should be *reactive*, sensing environment changes and behaving accordingly
- An agent should be *proactive*, deliberating upon its own course of actions based on its mental representation of the world
- So, more generally, how should the duality between reactivity and proactivity be ruled / balanced?

A DAI Agent Solves Problems

An agent has inferential capabilities

- New data representing a new solution to a given problem
- New knowledge inferred from old data
- New methods to solve a given problem
- New laws describing a portion of the world

An agent can change the world

- An agent is equipped with actuators that provide it with the ability to affect its environment
- The nature of actuators depends on the nature of the environment in which the agent is immersed / situated
- In any case, agent's ability to change the world is indeed limited

A DAI Agent Deliberates & Plans

An agent has a goal to pursue

- A *goal*, typically, as a state of the world to be reached—something to achieve
- A *task*, sometimes, as an activity to be brought to an end—something to do

An agent understands its own capabilities

- Its capabilities in terms of actions, pre-conditions on actions, effects of actions
- "Understands" roughly means that its admissible actions and related notions are somehow represented inside an agent, and there suitably interpreted and handled by the agent
- Perception should in some way interleave with action either to check action pre-conditions, or to verify action effects

An agent is able to build a plan of its actions

- It builds possible plans of action according to its goal/task, and to its knowledge of the environment
- It deliberates on the actual course of action to follow, then acts consequently

A DAI Agent is Flexible & Adaptable

Define *flexible*. Define *adaptable*.

- What do these words *exactly* mean?
- Adaptable / flexible with respect to what?
- Can an agent change its goal dynamically?
- Or, can it solve different problems in different contexts, or in dynamics contexts?
- Can an agent change its strategy dynamically?
- These properties are both important and potentially misleading, since they are apparently intuitive, and everybody thinks he/she understands them exactly

A DAI Agent Learns

What is (*not*) learning?

- Learning is not merely agent's change of state
- Learning is not merely dynamic perception—even though this change the agent's state and knowledge

What could an agent learn?

- New knowledge
- New laws of the world
- New inferential rules?
  - new ways to learn?

A number of areas insisting on this topic

Machine Learning, Abductive / Inductive Reasoning, Data Mining, Neural Networks, ...

DAI Agents: Summing Up

In the overall, a DAI agent has a number of important features

- It has a (partial) representation of the world (state & laws)
- It has a limited but dynamic perception of the world
- It has inferential capabilities
- It has a limited but well-known ability to change the world
- It has a goal to pursue (or, a task to do)
- It is able to plan its course of actions, and to deliberate on what to do actually
- Once understood what this means, it might also be flexible and adaptable
- It learns, regardless of how this term is understood

A PL Agent is Autonomous in Control

Complexity is in the control flow

- The need is to abstract away from control
- An agent encapsulates control flow
- An agent is an independent *locus* of control
- An agent is never invoked—it merely follows / drives its own control flow
- An agent is *autonomous* in control
  - it is never invoked—it cannot be invoked

A PL Agent is neither a Program, nor an Object

An agent is not merely a program

- A program represents the *only* flow of control
- An agent represents a single flow of control within a multiplicity

An agent is not merely a "grown-up" object

- An object is invoked, and simply responds
- An agent is never invoked, and can deliberate whether to respond or not to any stimulus

A P&D Agent is mobile [Fuggetta et al., 1998]

An agent is not bound to the Virtual Machine where it is born

- Reversing the perspective
  - it is not that agents are mobile
  - it is that objects are not
- *Mobility* is then another dimension of computing, just uncovered by agents

A new dimension requires new abstractions

- New models, technologies, methodologies
- To be used for reliability, limitations in bandwidth, fault-tolerance, ...

A Robotic Agent is Physical & Situated

A robot is a physical agent

- It has both a computational and a physical nature
  - complexity of physical world enters the agent boundaries, and cannot be confined within the environment

A robot is intrinsically situated

- Its intelligent behaviour cannot be considered as such separately from the environment where the robot lives and acts
- Some intelligent behaviour can be achieved even without any symbolic representation of the world
  - non-symbolic approach to intelligence, or *situated action* approach [Brooks, 1991]
- *Reactive architectures* come from here

A SE Agent is an Abstraction

An agent is an abstraction for engineering systems

- It encapsulate complexity in terms of
  - information / knowledge
  - control
  - goal / task
  - intelligence
  - mobility
- Agent-Oriented Software Engineering (AOSE)
  - engineering computational systems using agents
  - agent-based methodologies & tools

A MAS Agent is a Melting Pot

Putting everything together

- The area of Multiagent Systems (MAS) draws from the results of the many different areas contributing a coherent agent notion
- The MAS area is today an independent research field & scientific community
- As obvious, MAS emphasise the *multiplicity* of the agents composing a system

Summing up

- A MAS agent is an autonomous entity pursuing its goal / task by interacting with other agents as well as with its surrounding environment
- Its main features are
  - autonomy / proactivity
  - interactivity / reactivity / situatedness

A MAS Agent is Autonomous

A MAS agent is goal / task-oriented

- It encapsulates control
- Control is finalised to task / goal achievement

A MAS agent pursues its goal / task...

- ...proactively
- ...not in response to an external stimulus

So, what is new here?

- agents are goal / task oriented...
- ...but also MAS as wholes are
- *Individual* vs. *global* goal / task
  - how to make them coexist fruitfully, without clashes?

A MAS Agent is Interactive

Limited perception, limited capabilities

- It depends on other agents and external resources for the achievement of its goal / task
- It needs to interact with other agents and with the environment [Agre, 1995]
  - communication actions & pragmatical actions

A MAS agent lives not in isolation

- It lives within an agent *society*
- It lives immersed within an agent *environment*

Key-abstractions for MAS

- agents
- society
- environment

The Notion of Agent is Multi-faceted

Many reliable scientific sources

- Many more or less convergent / divergent definitions
- A synthesis is currently ongoing in the MAS community

Finally, defining the agent notion

- It is now possible...
- ...but it is also insufficient, now
- to fully define MAS

Meta-model is incomplete

- What about agent society?
- What about agent environment?

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

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

→ 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 intrinsically *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 artefacts 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 follows, too

- 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 *Exactly* a Goal

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

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

"Weak" Notion of Agent

Four key qualities [Wooldridge and Jennings, 1995]

Weak agents are

- Autonomous
- Proactive
- Reactive (to change)
- Social

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
- While intelligence is not mandatory for an agent to be autonomous
  - however, *intelligent autonomous agents* clearly make sense

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

"Strong" Notion of Agent

Mentalistic notion [Wooldridge and Jennings, 1995]

Strong agents have mental components such as

- Belief
- Desire
- Intention
- Knowledge
- ...

Intelligent agents and mental components

Intelligent autonomous agents are naturally (and quite typically) conceived as strong agents

Summing Up

Definition (Agent)

Agents are *autonomous computational entities*

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

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*

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Multi-, Inter- Trans-Disciplinary Research

From multi- to trans-disciplinary research

- Multi-disciplinary research means that multiple areas are involved in the same research activity—results are drawn from and concern different fields
- Inter-disciplinary research means that models, methods and techniques are brought from one area to a different one—results mainly concern the latter area
- Trans-disciplinary research means that models, methods and techniques are first brought from one area to a new one; then, once are suitably extended and generalised, results are brought back to the original area

Why X-Disciplinary Research?

Convergence of Scientific Research

- Complexity of systems (observed, modelled, constructed) is characterising more or less all of the human knowledge
- The same patterns in observable phenomena, system structure & behaviour, scientific models, methods, and techniques, occur repeatedly in many heterogeneous research fields

Convergence towards MAS

- Complexity of computational systems today matches complexity of biological, social, economical, organisational, . . . , systems
- Results from other areas dealing with complex systems may be useful / important / essential for computational systems & MAS in particular
- Results from computational systems & MAS are already changing the way in which scientific activity is conducted in every other areas

Questions to be Answered

We already learned something...

- ... about the reasons behind the agent abstraction,
- as well as some of its features

However, before a complete and precise definition could be given, some issues have to be clarified

We have to understand...

- ...if agents are the next thing after objects, what happens to objects, then? What about the paradigm shift?
- ...as object-oriented systems are made of interacting objects, are multiagent systems made of interacting agents—only?
- ...if societies and environment are essential to agent-oriented systems, how should they be handled in MAS modelling and engineering?
- ...if agents have to act, which are the objects of their acting?

Finally, we would like to taste...

- ...the flavour of X-disciplinary research

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Activity Theory (AT)

Origins of (Cultural-Historical) Activity Theory

- Born in the context of Soviet Psychology
- Rooted in the dialectic materialism by Marx & Engels
- Mostly by the work by Lev Vygotsky (1926-62) [Vygotskiĭ, 1978]
- Broadly speaking, AT is a very general framework for conceptualising human activities – how people learn, how society evolves – based on the concept of human *activity* as the fundamental unit of analysis

Activity Theory nowadays

- Re-discovered and widely applied in Computer Science and related fields in the last years [Nardi, 1996]
- Mostly in fields like Computer Supported Cooperative Work (CSCW) and Human Computer Interaction (HCI)
- Brought to the MAS field by both Italian and Spanish groups
  - e.g., [Ricci et al., 2003]

Human Activity in AT

Main Focus of AT

- AT focuses on *human activities*
- within a *social / organisational context*
- as separated by their respective (physical and ideal) *objects*

Collaborative activities in AT

- Cooperation is understood as a *collaborative* activity
- A collaborative activity has *one objective*
- A collaborative activity is *distributed* onto *several actors*, who participate to the activity
- Explicit norms and rules regulate the relationships among individual participants' work

Mediated Interaction in AT

Every Human Activity...

- ...is found to be *mediated*...
- ...by mediating *artefacts*...
- ...of heterogeneous nature, either physical or psychological
  - operating procedures, heuristics, scripts, languages, ...

Artefacts in AT

- Artefacts are the tools that mediate actions and social interactions
  - artefacts mediate between individual participants and their *environment*
  - artefacts embody the portion of the environment that can be designed and controlled to support participants' activities
- As an observable part of the environment, artefacts can be monitored along with the development of the activities
  - to evaluate overall system performance and
  - to keep track of system history

Role of Artefacts in AT

- Artefacts can be either *physical* or *cognitive*—or, they may have a twofold nature
  - example of physical artefacts are shelves, doors, phones, whiteboards, ...
  - example of cognitive artefacts are operating procedures, heuristics, scripts, languages, ...
  - examples of artefacts with a twofold nature (physical / cognitive) are operating manuals, computers, ...
- Artefacts are both a *means* but also a *product* of social activity, so they embody a set of social practise
  - their design and structure reflect a history of particular use in some given social / organisational context

Artefacts as Enablers and Constrainers of Activities

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

enablers artefacts expand out possibilities to manipulate and transform different objects

constrainers the object is perceived and manipulated through the artefact not 'as such' but within the limitations set by the artefact itself

- A simple example: a driving wheel

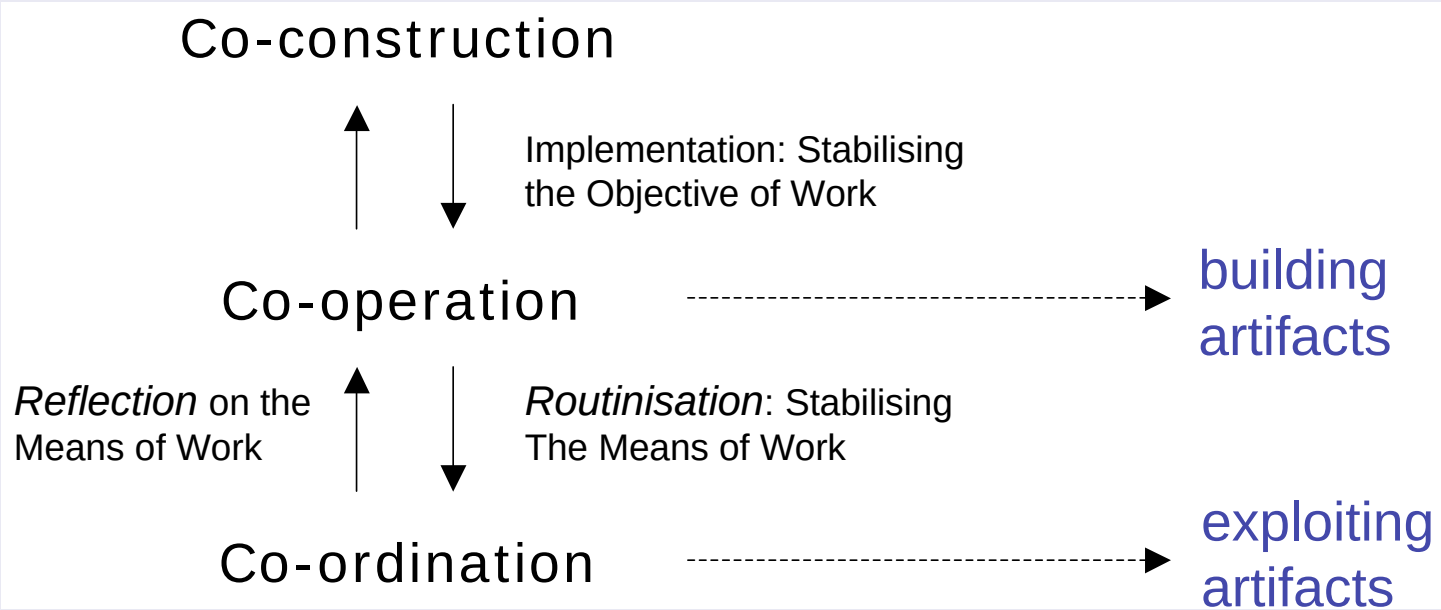
enabler enables me to change direction while driving a car

constrainer allows me only one way to change direction while driving a car

Layers for Collaboration Activities in AT

- AT identifies a three-layered structure for social (collaborative) activities [Bardram, 1998, Engeström et al., 1997]
- The three layers are labelled as
  - *co-ordinated*
  - *co-operative*
  - *co-constructive*

AT Layers: The Picture



Co-ordination in AT

- The *co-ordinated* aspect of work captures the *normal* and routine *flow of interaction*
- Participants follow their *scripted roles*, each focusing on the successful performance of their actions, implicitly or explicitly assigned to them
- Participants share and act upon a common object, but their individual actions are only externally related to each other
- *Scripts* coordinating participants' actions are not questioned or discussed, neither known and understood in all their complexity
- Participants act as "wheels in the organisational machinery" [Kuutti, 1991], and co-ordination ensures that an activity is working in harmony with surrounding activities

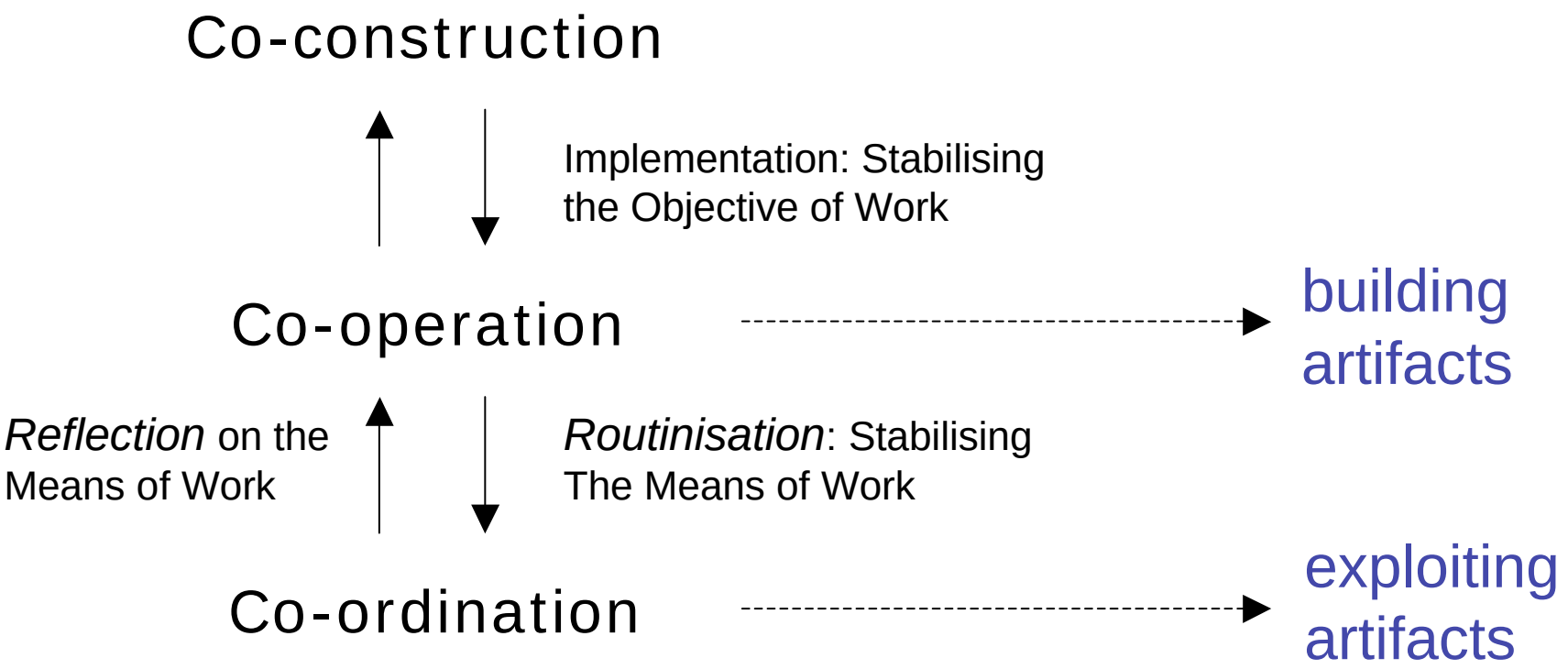
Co-operation in AT

- The *co-operative* aspect of work concerns the mode of interactions in which actors focus on a common object, thus *share* the *objective* of the activity
- Here, actors do not have actions or roles explicitly assigned to them
- With regard to the common object, each actor has to balance his/her own actions with other agent actions, possibly influencing them to achieve the common task
- At the co-operation level
  - the object of the activity is stable and agreed upon
  - the means for realising the activity is not yet defined
- The means for realising a collaborative activity—the artefacts—are then the *object* of the co-operative activity, and its results as well

Co-construction in AT

- The *co-constructive* aspect of work concerns interactions in which actors focus on re-conceptualising their own organisation and interaction in relation to their shared objects
- Neither the object of work, nor the scripts are stable, and must be collectively constructed, i.e., *co*-constructed

AT Layers: Summing Up



Collaborative activities in AT

- A collaborative activity is not to be seen in general at one single level
- Co-ordination, co-operation, and co-construction are instead to be interpreted as *analytical* distinctions of the same collaborative activity, concurring in different times and modes to its development

Agents are not the Only Abstractions Needed

Basic Abstractions: Agents plus Artefacts

- Adopting AT as a conceptual framework for MAS social activities has led to recognise that *agents are not the only basic abstractions* to model and build MAS [Ricci et al., 2003]
- Artefacts, too, are necessary [Ricci et al., 2006]
  - to enable and constrain agent actions
  - to mediate agent interactions with other agents and with the environment
  - to model and shape MAS *environment*
  - in general, to improve agent ability to achieve their individual and social goals

Relevance of AT Research in MAS

Artefacts are essential—in MAS, too

- AT investigation is relevant in MAS since it points out that artefacts are essential to enable and govern agent actions and interactions within a MAS
  - by enhancing agent capabilities to act
  - by constraining both individual and social activities in a MAS

Role of environment

- AT emphasises the fundamental role of the *environment* in the development of complex systems
- Also, AT suggests that artefacts are the essential tools [Weyns et al., 2007, Viroli et al., 2005]
  - to model MAS environment
  - to shape it so as to make it favourable to the development of collaborative activities

Coordination Artefacts

Artefacts for collaboration and coordination

- *Coordination artefacts* are artefacts used in the context of collaborative activities, mediating the interaction among actors involved in the same social context [Ricci et al., 2003]
- Coordination artefacts can be either *embodied* or *disembodied*, referring to respectively physically or cognitive/psychological artefacts
- Coordination artefacts are social artefacts shared by agents in a MAS, which are meant to enable and govern the interaction among agents, and between agents and their environment

Coordination artefacts & media

- Coordination artefacts represent a straightforward generalisation of the notion of coordination medium, as coming from fields like coordination models and languages and distributed AI
- Examples include abstractions like tuple spaces, channels, blackboards, but also pheromone infrastructures, e-institutions, ...

AT Layers for MAS Collaboration

Layers for MAS collaboration & coordination artefacts

- The three levels identified by AT for social activities can be re-interpreted in the MAS context in terms of the relationship between agents and artefacts—in particular, coordination artefacts
- The three layers are labelled as
  - *co-ordination*
  - *co-operation*
  - *co-construction*

AT Layers for MAS in Detail

co-construction — agents understand and reason about the (social) objectives (goals) of the MAS, and build up a model of the social tasks required to achieve them—this also involves identifying interdependencies and interactions to be faced and managed

co-operation — agents design and build the coordination artefacts—either embodied (coordination media) or disembodied (plans, interaction protocols, etc.)—which are useful to carry on the social tasks and to manage the interdependencies and interactions devised out at the previous (co-construction) stage

co-ordination — agents use the coordination artefacts: then, the activities meant at managing interdependencies and interactions—either designed a-priori or planned at the co-operation stage—are enforced/automated

Levels of Use of Artefacts

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

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

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

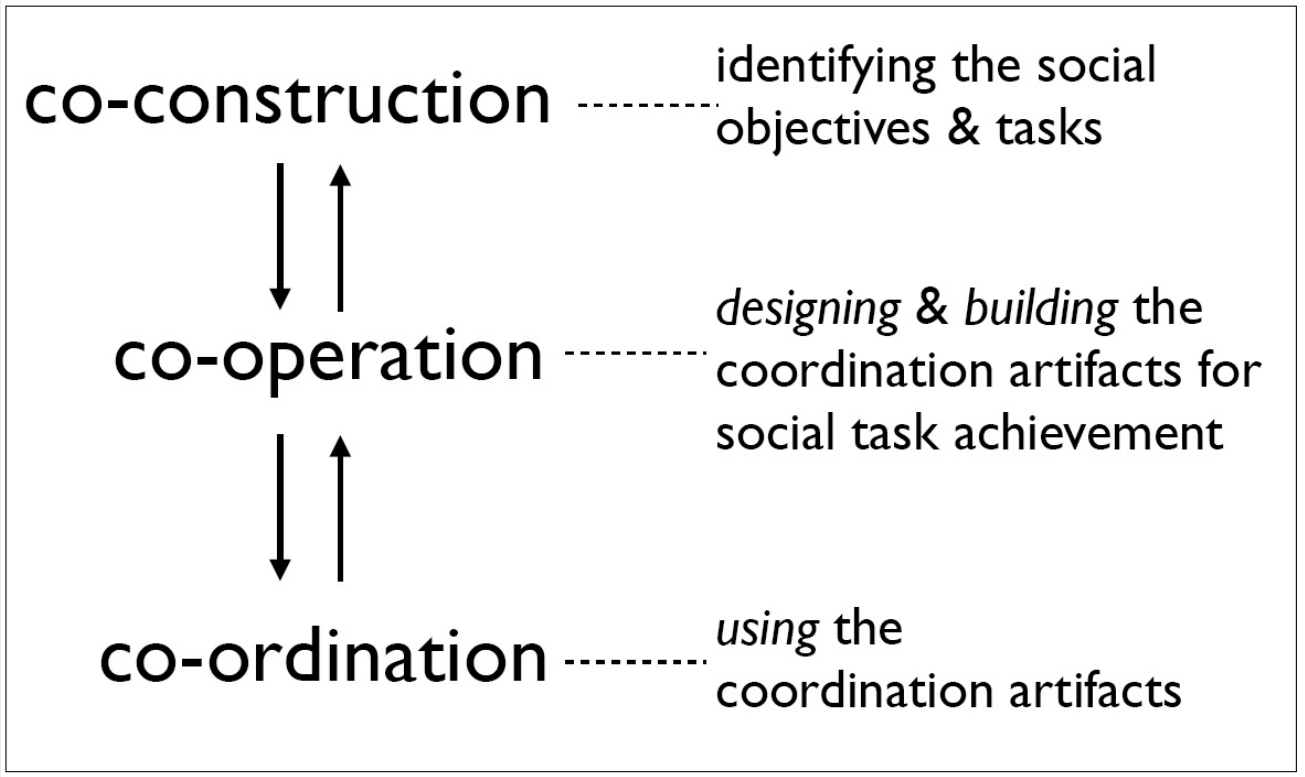
Intelligent agents could possibly reason about the nature of the artefacts as well as on the level of achievement of their goals, and take the chance to change or adapt the artefacts, or even to create new ones whenever useful and possible as the result of either an individual or a social activity

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

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

AT Layers for MAS Collaboration: The Picture

AT Layers for MAS: The Picture



Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Distributed Cognition in Short

Distributed Cognition. . . [Kirsh, 1999]

- . . . is a branch of cognitive sciences
- which proposes that *human cognition* and *knowledge representation*
  - rather than solely confined within individuals
  - is *distributed* across individuals, tools and *artefacts* in the *environment*

Cognition is Distributed

Cognition transcends individuals

- Intelligent processes in human activity go beyond the boundaries of individual actors
- Knowledge is not confined within human minds
- Cognition transcends individual cognition

Knowledge representation transcends individuals

- Knowledge representation does not pertain individual humans only
- Representation is distributed
  - partially in the mental spaces of humans
  - as external representations of memories, facts, and information of any sort distributed on the objects, tools and instruments that constitute the environment

Distributed Cognitive Systems

Analysis of Distributed Cognition focuses on *distributed cognitive systems*

- People interact with external *cognitive artefacts* containing knowledge represented in some form
- Human *intelligent behaviour* results from the distributed interactions with other humans and with cognitive artefacts
- In the overall, this defines and determines the context where human activities are *situated*
  - that is, the physical, cultural and social context that also guides, constrains and partially determines intelligent activities

Cognitive Artefacts

Cognitive artefacts: a definition [Norman, 1992]

those artificial devices that maintain, display, or operate upon information in order to serve a representational function and that affect human cognitive performance

Cognitive artefacts are...

- ...a product of human design and work
- ...aimed at aiding or enhancing our cognitive abilities
  - like post-its, calendars, agendas, computers, etc.
- ...not mere amplifiers of our cognitive abilities
  - cognitive artefacts also modify the nature of the tasks to be performed

Personal vs. System View

System view

Individuals plus artefacts altogether as a (functional) subsystems

- Understanding activities requires to consider (cognitive) actors and (cognitive) artefacts altogether
- Actions are sometimes mediated sometimes targeted to artefacts, and cannot be fully understood without them

Personal view

Individuals as subsystems affected by artefacts

- Practical reasoning is deeply affected by artefacts
- Individuals should change the way in which they represent actions, plan, deliberate and finally act

Environment in Distributed Cognitive Systems

Environment has a key role in distributed cognitive systems

- In distributed cognitive systems, the nature of the environment
  - on the one hand, depends on the artefacts and tools that shape it
  - on the other hand, determines the efficiency and effectiveness of the work and activities of the actors that are immersed in it

Work environment

- How do we define a working environment for individuals and organisations?
  - it mostly depends on the tasks that have to be carried on inside
- Real work environments are a complex superposition of social, cultural, cognitive, and physical constraints
- How should the environment be understood as a complex analytical construct when the goal is environment design?

Coordination in Distributed Cognitive Systems

Observing real world activities

- An effective environment for a successful activity is a shifting coalition of resources and constraints
  - some physical, some social, some cultural, some computational
  - involving both internal and external computational resources
- Activity is successful whenever such a coalition is suitably coordinated
  - lack of coordination means failure of activity
- Coordination is then essential, and concerns activities, resources and constraints
  - at both the individual and the social level

The Function of Action in Distributed Cognitive Systems

What is the purpose of an activity?

- A dominant assumption is that the point of activity is to change the environment in a way that (presumably) leads to goal satisfaction
- Many action however do not make sense under this assumption
  - most communication actions, but not only them
  - for instance, people undertake actions to save attention, memory and computation; people recruit external elements to reduce their own cognitive effort by distributing computational load
  - this make sense if people is situated
- As a result, environment design should not merely be aimed at helping people to achieve their goals
  - it should also be designed to make other actions easy
  - such as epistemic, complementary, coordinative actions

Cognition Outside Agents

Cognition & knowledge representation do not belong to agents only

- Objects & tools in the environment may participate to the cognitive processes
- Structure of MAS environment may explicitly represent knowledge

Cognition & knowledge representation are distributed in the environment

- Artefacts are essential parts of the MAS cognitive processes
- Cognitive artefacts encapsulate knowledge as explicitly represented

Agent View vs. MAS View

Personal / agent view

- Once artefacts are exploited, they change the way in which agents act and reason about action

System / MAS view

- In order to understand and possibly evaluate agent (social) action within a MAS, one should consider agent(s)+artefact(s) altogether

MAS Environment is Structured

(Cognitive) artefacts shape MAS environment

- Artefacts determine the structure of MAS environment
- Knowledge is distributed in the environment, and encapsulated within cognitive artefacts
- Structure of the environment, and knowledge it contains, affect the activities of agents within MAS

MAS Action & Coordination

MAS coordination depends on environment structure

- Environment structure changes the nature of agent action
- Environment structure affects agent mutual interaction
- Environment structure modifies the way agents coordinate in a MAS
- Environment structure should be designed to
  - help agent actions to achieve their goals
  - help epistemic, complementary, coordinative agent actions easier / effective

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Agents & Goals

Cognitive interpretation of (social) action [Conte and Castelfranchi, 1995]

- Agents in a society can be generally conceived as either *goal-governed* or *goal-oriented* entities
  - goal-governed entities refer to the strong notion of agency, i.e. agents with some forms of cognitive capabilities, which make it possible to explicitly represent their goals, driving the selection of agent actions
  - goal-oriented entities refer to the weak notion of agency, i.e. agents whose behaviour is directly designed and programmed to achieve some goal, which is not explicitly represented
- In both cases, agent goals are *internal*

External goals

- *External goals* refer to goals that typically belong of the social context or *environment* where the agents are situated
- External goals are sorts of regulatory states which condition agent behaviour
  - a goal-governed system follows external goals by adjusting internal ones

Entities without Goals

Not every entity involved in (social) actions has a goal

- Within a society, there are entities that are explicitly designed to provide a *function*
- *Artefacts* are such objects
  - they have a function associated
- Artefacts have no goals to achieve
  - they may have a *destination* associated
    - a *destination* is external goal attached to an artefact by an agent, in the act of using it
  - destination is then associated to the *use* of an artefact
    - destination is related but not identical to function: an artefact can be used according to a destination that differs from its designed function

On the Relation Between Agents & Artefacts [Conte and Castelfranchi, 1995]

Use & use value

- When facing an artefact, an agent may adopt different perspectives
- Evaluating an artefact for use, to select it among many others, and then to use it, to achieve agent's own goals, are two different matters
- Different sorts of external goals are associated by an agent to an artefact

use value the *use-value* goal, according to which the artefact should allow user agents to achieve their objective—this drives the agent *selection* of the artefact

use the *use* goal, which directly corresponds to the agent internal goal—this guides the actual *usage* of the artefact

Articulating the Agents & Artefacts Relationship

How could an agent deal with an artefact?

- There are at least three different ways an agent can exploit an artefact

use by merely using it, according to its function, and associating it to a destination

selection by selecting it for future use, according to its function, its possible future destinations, and the agent's goals and plans

construction & manipulation by adapting & changing an existing artefact, or by creating a new one for future use, thus designing its function, according to its possible future destinations, and the agent's goals and plans

Goals in MAS

Agents have goals

strong agency Agents have explicitly-represented goals

weak agency Agents have implicitly-represented / encoded goals

Artefacts have functions

- Artefacts have no internal goals
- Artefacts have a pre-designed function
- An artefact is associated with an external goal (its destination) by agents in the act of using it

Agents & Artefacts Interacting

Aspects of agent-artefact relationship

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

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

selection An agent can select an artefact 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 artefact 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 artefact with an agent-designed function according to some required use-value goals and to its possible future destinations

MAS Engineers Designing Agents

Basic choices to make in agent design

- Should an agent be aware of artefact's behaviour and structure, and of how to use them?
  - should an agent be able to reason and deliberate about artefact use?
- Should an agent be aware of artefact's function and possible uses?
- Should an agent be able to act over artefacts to modify them and adapt their function?
  - should an agent be able to create *ad hoc* artefacts *ex novo*?
- Should a MAS engineer be able to act over artefacts to modify them and adapt their function, or, to create new artefacts, *at run-time*?

MAS Engineers Designing Artefacts

Basic issues in artefact design

- How should an artefact be made in order to be ready for agent's use?
  - either aware, or unaware
  - possibly, within an open system
- How should an artefact be made in order to be ready for agent's evaluation and selection?
- How should an artefact be made in order to be ready for agent's modification and adaptation?
- How should MAS environment be structured in order to allow artefact run-time creation and modification?
  - by agents and MAS engineers?

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Computer Supported Cooperative Work (CSCW)

Basic issues in CSCW

- CSCW aims at automating human cooperative work through computational procedures
- However, two diverging strategies are currently emerging [Schmidt and Simone, 2000]

  automation stressing computational procedures to automate coordination of activities

  flexibility stressing the flexibility of computational procedures with respect to intelligent coordination by collaborating actors

- The former approach emphasises coordination by the computational entities ruling collaboration, the latter coordination by intelligent collaboration entities
- Main problem: the two strategies *diverge*, they should instead *converge*

Automation vs. Flexibility: Key Issues in CSCW

Mutual awareness for flexibility

- *Mutual awareness* means that the actors of a collaboration activity affect and mutually perceive the other actor's activities through the shared workspace
- The so-called *common field of work* can reveal / conceal portions of the collaboration activities to the participants
- Mutual awareness is then the basis for opportunistic, *ad hoc* alignment and improvisation, which ensure *flexibility* to collaborative activities

Coordinative artefacts for automation

- *Coordinative artefacts* are the rulers of collaboration
- They work more as *constrainers* rather than as *commanders*
- By giving structure to the common field of work, coordinative artefacts encapsulate those coordination responsibilities that are better to be *automatised* in order to achieve efficiency in cooperation
- In all, coordinative artefacts

  work as constrainers they define and govern the space of the admissible articulation of activities

  work *not* as commanders they do not impose a pre-defined course of actions that could cause unnecessary rigidity and reduce the required flexibility

Automation of Collaboration Activities in MAS

Coordinative artefacts for automation of MAS collaboration

- *Coordinative artefacts* rule MAS collaboration, working more as constrainers rather than as commanders
- Coordinative artefacts structure MAS common field of work, as specialised abstractions automatising and making collaboration efficient
- As constrainers, coordinative artefacts define and govern the space of the admissible articulation of MAS collaboration activities
- On the other hand, they do not impose a pre-defined course of actions, promoting flexibility of intelligent agent coordination, and respecting agent autonomy

Flexibility of Collaboration Activities in MAS

Mutual awareness for flexibility of MAS collaboration

- Shared MAS environment should be structured as the MAS *common field of work* to allow agents to mutually perceive each other's activities (*mutual awareness*)
- MAS common field of work can reveal / conceal portions of MAS collaboration activities to the agents
- Mutual awareness promotes opportunistic alignment and improvisation of agent activities, and ensure *flexibility* to MAS collaboration

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Anthropology: The Logocentric Philosophical Bias

Human capacity of language as the main sign of intelligence?

- Western anthropology has long dwelt on such a point
- The relation between language, use of tools, and evolution of intelligence has long been neglected [Hewes, 1993]

Human capacity of developing and using tools as a fundamental sign of intelligence

- Humans forge & use tools
- The first characterisation of *Homo Abilis* is its ability to forge tools
  - tool using vs. tool making distinction
- This is a clear sign of intelligence
- Evidence of co-evolution of language and tools use along with human intelligence is overwhelming in modern anthropological studies [Gibson and Ingold, 1993]

Ethology: Tools, Animals, and the Tooling Test

Use of tools is not an exclusive feature of humans

- Beavers build dams, bees construct perfect hexagonal cells, many birds live in self-made nests, ...
- Ethologists commonly measure intelligence of animals by making them face problems that require the use of tools to be solved (e.g. [Povinelli, 2000])
- A sort of tool-equivalent of the Turing test has been proposed by philosopher Ronald Endicott, which was aimed at evaluating intelligence in terms of the ability to exploit tools
  - the so-called "Tooling Test for Intelligence" [Wood et al., 2005]

Anthropology from Theologists: Tools & Self-Awareness

Use of tools reveals awareness of self [Martelet, 1998]

- When using a tool, a creature shows it is able to distinguish and identify itself from the world around
- The use of a tool *reveals awareness* of self, and of the environment as well
  - whenever a tool is built with a goal, it is stored for further / repeated use, it is used for building new tools, etc.
- Tools are at the same time the first and the most distinctive *expression of human intelligence*, along with language
- They are the most powerful amplifiers of the (both individual and social) *human ability to affect the environment*—to survive environment change, first, and to change the environment for the human purposes, then

The Logocentric Philosophical Bias in MAS

Agent capacity of language as the main sign of agent intelligence?

- Research on MAS still dwells on the *logocentric bias*
- Intelligent *use of tools by agents* is typically neglected
  - as a stunning example, FIPA (Foundation for Intelligent Physical Agents) just ignore pragmatic / physical agent actions, and only focuses on agent communication actions

Agent ability of developing and using tools as a sign of agent intelligence

- A notion of *tool* for agents is needed
- Agents should be *able to use tools*
- Intelligent agents should be *able to forge & adapt tools*
- A theory of physical / pragmatical action should be developed for agents, as refined as the one for communication actions
- Such a theory should focus on tool use / creation by agents
  - The notion of *Agens Faber* goes along this very direction [Omicini et al., 2006]

Tools, Agents, and the Tooling Test

Use of tools should be a feature for agents in a MAS

- MAS researchers should be able to measure intelligence of agents by making them face problems that require the use of tools to be solved
- A sort of tool-equivalent of the Turing test for agents using tools should be defined, aimed at evaluating agent intelligence in terms of the ability to exploit tools
  - a sort of "Tooling Test for Agent Intelligence"
- Agent intelligence should then be measured by both the agent ability to communicate and by agent ability to use tools
  - the two abilities should be somehow strictly related, and "co-evolve" in some sense—a common theory of agent action could be of use here

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

The A&A Meta-model

A&A: A conceptual framework for MAS modelling & engineering

Based on the conceptual foundations discussed in the previous block of slides, the A&A meta-model is a conceptual framework characterised in terms of three basic abstractions [Omicini et al., 2008]:

agents represent pro-active components of the systems, encapsulating the autonomous execution of some kind of activities inside some sort of environment

artefacts represent passive components of the systems such as resources and media that are intentionally constructed, shared, manipulated and used by agents to support their activities, either cooperatively or competitively

workspaces are the conceptual containers of agents and artefacts, useful for defining the topology for the environment and providing a way to define a notion of locality

Reminder: 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*

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Artefacts in the A&A Meta-model

Definition (A&A Artefact)

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

genus artefacts are computational entities

differentia artefacts are aimed to be used by agents

Artefacts 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

Artefacts Have a Function

Artefacts are designed for use

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

Artefacts are transparent & predictable

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

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

Artefacts Are Not Autonomous

Artefacts are designed to serve

- Artefacts are designed to *serve* some agent's purpose
  - not to follow their own path of action
- An artefact has an embodied function, made repeatedly and predictably available to agents
- An artefact 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

Artefacts Are (Computationally) Reactive

Artefacts are *reactive* in terms of control

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

Artefacts Have Operations and Interfaces

Agents use artefact operations

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

Artefacts are Situated

Artefacts & Agent Actions

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

Artefacts are reactive to change

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

Artefacts Are Not Agents

Agents vs. artefacts

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

Artefacts in the A&A Meta-model

Definition (A&A Artefact)

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

genus artefacts are computational entities

differentia artefacts are aimed to be used by agents

Artefacts are *to be used* by agents

- From use, many other features stem
  - artefacts 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
  - artefacts are not agents

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Artefacts & Environment

Artefacts as mediators

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

Artefacts as representatives of MAS environment

- As an observable & controllable part of the environment, artefacts 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

Artefacts for environment design

- Artefacts 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

Artefacts as Enablers and Constrainers of MAS Activities

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

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

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

How do we like artefacts?

- Artefacts 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 Artefacts: Inspectability

- The state of an artefact, its content (whatever this means in a specific artefact), 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 artefact 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 artefacts
- Also, artefacts are often in charge of critical MAS behaviour [Omicini et al., 2004a]: being able to inspect a part or the whole of an artefact 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 Artefacts: Controllability

- *Controllability* is an obvious extension of the inspectability property
- The operational behaviour of an artefact 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 artefact 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 artefact at execution time, and of the associated MAS social activities as well

A&A Artefacts: Malleability

- Also related to inspectability, *malleability* (also called *forgeability*) is a key-feature in dynamic MAS scenarios, when the behaviour of artefacts 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 artefact 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 Artefacts: Predictability

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

A&A Artefacts: Formalisability

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

A&A Artefacts: Linkability

- Artefacts 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 artefacts, e.g. to build a service incrementally on top of another, by making a new artefact realising its service by interacting with an existing artefact
- To this end, artefacts should be able to invoke the operation of another artefact: the reply to that invocation will be transmitted by the receiver through the invocation of another operation upon the caller

A&A Artefacts: Distribution

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

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

Levels of Use of Artefacts

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

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

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

Intelligent agents could possibly reason about the nature of the artefacts as well as on the level of achievement of their goals, and take the chance to change or adapt the artefacts, 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 artefacts

In the same way, MAS engineers can use artefacts 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 & Artefacts Interacting

Aspects of agent-artefact relationship

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

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

selection An agent can select an artefact 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 artefact 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 artefact with an agent-designed function according to some required use-value goals and to its possible future destinations

Cognitional Artefacts

Definition (A&A Cognitional Artefact)

An A&A cognitional artefact is an artefact *aimed* at the *cognitive use* by agents

genus cognitional artefacts are artefacts

differentia cognitional artefacts are aimed to be used in a cognitive way by agents

Rational exploitation of (cognitional) artefacts by cognitive agents

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

Cognitional Artefacts: Usage Interface

Agents, artefacts & operations

- One of the core differences between artefacts and agents is the concept of *operation*
- An operation is the means by which an artefact provides agents with a service or function
- An agent executes an action over an artefact by invoking an artefact 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 artefact defines what is called its *usage interface*

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

Cognitional Artefacts: Operating Instructions

Artefact's manuals for intelligent agents

- Operations cannot be invoked in any order
- Artefact's state & behaviour, along with the effects of agent's actions on the environment via the artefact, 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 artefact 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 artefact, 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
- Artefacts 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

Cognitional Artefacts: Function Description

Agents, artefacts & function

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

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

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

An example

When modelling a sensor wrapper as an artefact, 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

Outline

1. Agents for Complex Software Systems
   - Toward a Paradigm Change
   - Away from Objects
   - The Many Agents Around
2. Agents: Definitions & Conceptual Framework
   - Autonomy
   - Definitions
3. Artefacts: Multi-disciplinary Foundations
   - Premises
   - Activity Theory
   - Distributed Cognition
   - Sociology
   - Computer Supported Cooperative Work
   - (Cognitive) Anthropology & Ethology
4. Agents & Artefacts: Definitions & Conceptual Framework
   - Agents & the A&A Meta-model
   - On the Notion of Artefact in the A&A Meta-model
   - MAS Engineering with A&A Artefacts
   - A&A Artefacts for Cognitive Agents
   - On the Notion of MAS in the A&A Meta-model

MAS in the A&A Meta-model

Definition (A&A MAS)

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

genus MAS is computational system

differentia its basic components are agents and artefacts

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 & artefacts
- Both agents & artefacts 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 & artefacts
- Agents are pro-active, artefacts are reactive
- Agents are autonomous entities, artefacts have functions

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

MAS Interaction in the A&A Meta-model

Admissible interactions *within* a MAS

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

communication agents *speak* with agents
operation agents *use* artefacts
composition artefacts *link* with artefacts
presentation artefacts *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 & artefacts, 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 artefact 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

Agre, P. E. (1995).
Computational research on interaction and agency.
Artificial Intelligence, 72(1-2):1–52.
Special volume on computational research on interaction and agency, part 1.

Bardram, J. (1998).
Designing for the dynamics of cooperative work activities.
In *1998 ACM Conference on Computer Supported Cooperative Work (CSCW'98)*, pages 89–98. ACM Press.

Brooks, R. A. (1991).
Intelligence without representation.
Artificial Intelligence, 47:139–159.

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.

Conte, R. and Castelfranchi, C., editors (1995).
Cognitive and Social Action.
Routledge.

Bibliography II

Engeström, Y., Brown, K., Christopher, L. C., and Gregory, J. (1997).
Coordination, cooperation, and communication in the Courts: Expansive transitions in legal work.
In Cole, M., Engeström, Y., and Vasquez, O. A., editors, *Mind, Culture, and Activity. Seminal Papers from the Laboratory of Comparative Human Cognition*, chapter 28, pages 369–388. Cambridge University Press.

Fuggetta, A., Picco, G. P., and Vigna, G. (1998).
Understanding code mobility.
IEEE Transactions on Software Engineering, 24(5):342–361.

Gibson, K. R. and Ingold, T., editors (1993).
Tools, Language and Cognition in Human Evolution.
Cambridge University Press.

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.

Bibliography III

Hewes, G. W. (1993).
A history of speculation on the relation between tools and languages.
In [Gibson and Ingold, 1993], pages 20–31.

Kirsh, D. (1999).
Distributed cognition, coordination and environment design.
In Bagnara, S., editor, *3rd European Conference on Cognitive Science (ECCS'99)*, pages 1–11, Certosa di Pontignano, Siena, Italy. Istituto di Psicologia, Consiglio Nazionale delle Ricerche.

Kuhn, T. S. (1996).
The Structure of Scientific Revolutions.
University of Chicago Press, 3rd edition.

Kuutti, K. (1991).
The concept of activity as a basic unit of analysis for CSCW research.
In Bannon, L. J., Robinson, M., and Schmidt, K., editors, *2nd European Conference on CSCW (ECSCW'91)*, pages 249–264. Kluwer Academic Publisher.

Martelet, G. (1998).
Évolution et création, tome 1 – Sens ou non-sens de l'homme dans la nature?
Editions du Cerf, Paris.

Bibliography IV

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.

Nardi, B. A., editor (1996).
Context and Consciousness: Activity Theory and Human-Computer Interaction.
MIT Press.

Norman, D. A. (1992).
Design principles for cognitive artifacts.
Research in Engineering Design, 4(1):43–50.

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

O'Hare, G. M. and Jennings, N. R., editors (1996).
Foundations of Distributed Artificial Intelligence.
Sixth-Generation Computer Technology. John Wiley & Sons Ltd., hardcover edition.

Bibliography V

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. and Poggi, A. (2006).
Multiagent systems.
Intelligenza Artificiale, III(1-2):76–83.
Special Issue: The First 50 Years of Artificial Intelligence.

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.

Bibliography VI

Omicini, A., Ricci, A., and Viroli, M. (2008).
Artifacts in the A&A meta-model for multi-agent systems.
Autonomous Agents and Multi-Agent Systems, 17(3):432–456.
Special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems.

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.

Povinelli, D. J. (2000).
Folk Physics for Apes: The Chimpanzee's Theory of How the World Works.
Oxford University Press.

Ricci, A., Omicini, A., and Denti, E. (2003).
Activity Theory as a framework for MAS coordination.
In Petta, P., Tolksdorf, R., and Zambonelli, F., editors, *Engineering Societies in the Agents World III*, volume 2577 of *LNCS*, pages 96–110. Springer-Verlag.
3rd International Workshop (ESAW 2002), Madrid, Spain, 16–17 September 2002. Revised Papers.

Bibliography VII

Ricci, A., Viroli, M., and Omicini, A. (2006).
Programming MAS with artifacts.
In Bordini, R. P., Dastani, M., Dix, J., and El Fallah Seghrouchni, A., editors, *Programming Multi-Agent Systems*, volume 3862 of *LNAI*, pages 206–221. Springer.
3rd International Workshop (PROMAS 2005), AAMAS 2005, Utrecht, The Netherlands, 26 July 2005. Revised and Invited Papers.

Russell, S. J. and Norvig, P. (2002).
Artificial Intelligence: A Modern Approach.
Prentice Hall / Pearson Education International, Englewood Cliffs, NJ, USA, 2nd edition.

Schmidt, K. and Simone, C. (2000).
Mind the gap! Towards a unified view of CSCW.
In Dieng, R., Giboin, A., Karsenty, L., and De Michelis, G., editors, *Designing Cooperative Systems: The Use of Theories and Models*, volume 58 of *Frontiers in Artificial Intelligence and Applications*, Sophia Antipolis, France. IOS Press.
4th International Conference on the Design of Cooperative Systems (COOP 2000), Proceedings.

Bibliography VIII

Viroli, M., Omicini, A., and Ricci, A. (2005).
Engineering MAS environment with artifacts.
In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *2nd International Workshop "Environments for Multi-Agent Systems" (E4MAS 2005)*, pages 62–77, AAMAS 2005, Utrecht, The Netherlands.

Vygotskiĭ, L. S. (1978).
Mind in Society: Development of Higher Psychological Processes.
Harvard University Press.

Wegner, P. (1997).
Why interaction is more powerful than algorithms.
Communications of the ACM, 40(5):80–91.

Weyns, D., Omicini, A., and Odell, J. (2007).
Environment as a first-class abstraction in multi-agent systems.
Autonomous Agents and Multi-Agent Systems, 14(1):5–30.
Special Issue on Environments for Multi-agent Systems.

Wood, A. B., Horton, T. E., and Amant, R. S. (2005).
Effective tool use in a habile agent.
In Bass, E. J., editor, *2005 IEEE Systems and Information Engineering Design Symposium (SEADS 2005)*, pages 75–81, Charlottesville, VA, USA. IEEE.

Bibliography IX

Wooldridge, M. and Jennings, N. R. (1995).
Intelligent agents: Theory and practice.
Knowledge Engineering Review, 10(2):115–152.

Wooldridge, M. J. (2002).
An Introduction to MultiAgent Systems.
John Wiley & Sons Ltd., Chichester, UK.

Zambonelli, F. and Parunak, H. V. D. (2003).
Towards a paradigm change in computer science and software engineering: A synthesis.
The Knowledge Engineering Review, 18(4):329–342.

Computational Agents & Artefacts

Dottorato di Ricerca in Informatica Giuridica e Diritto dell'Informatica

Andrea Omicini
andrea.omicini@unibo.it

CIRSFID
Alma Mater Studiorum—Università di Bologna

Academic Year 2009/2010