Frontiers of Autonomous Systems

Andrea Omicini
andreaomicini.apice.unibo.it
andrea.omicini@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum—Università di Bologna, Italy

Collegio Superiore
Bologna, Italy
March 2018

Outline of Part I. Autonomy of Artificial Systems

Outline of Part II. System Autonomy & Self-Organisation

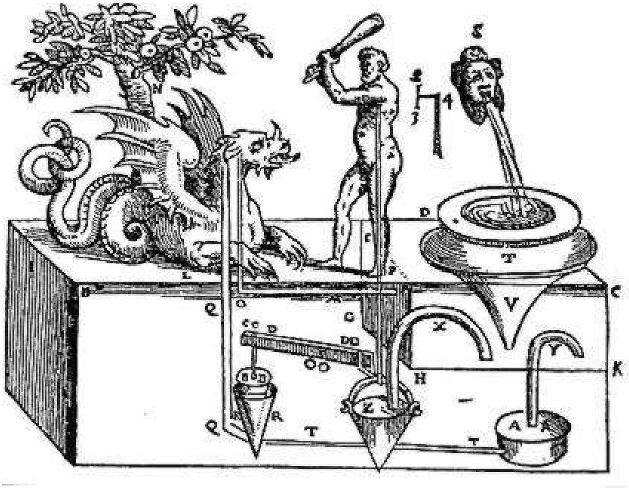
Part I

Autonomy of Artificial Systems

Next in Line. . .

Focus on. . .

The Obsession with (Autonomous) Machines I



The Obsession with (Autonomous) Machines II

Machines doing something "by themselves"

- an obsession coming with *technique*
  - basically, representing our way to *affect* the world around us
  - possibly, according to our *goals*
- China, Greece, Italy, England, France
  - hundred years of attempts, and some success, too

The Obsession with (Autonomous) Machines III

The Obsession with (Autonomous) Machines IV

The Obsession with (Autonomous) Machines V

Aren't humans just "playing God"?

- maybe, ok, good point.
- fascination was so huge, too
- so strong, that *fake* automata were even quite frequent, and even famous

The Obsession with (Autonomous) Machines VI

The Obsession with (Autonomous) Machines VII

The original question

- what can human artefacts actually do?
- what can they achieve?
- what can humans achieve with the systems they create?

Before Autonomous Systems: Machines I

Constructing for understanding

- building machines with
  - initiative
  - autonomy
  - knowledge

  for *understanding ourselves*, and the *world* where we live
- “playing God” to understand the world

Before Autonomous Systems: Machines II

Relieving humans from fatigue

- goal: substituting human work in
  - quality
  - quantity
  - cost
- more, better, cheaper work done
  - new activities become feasible
- which work?
  - first, physical
  - then, repetitive, enduring
  - subsequently, intellectual, too
  - finally, simply more complex for any reason—or, all reasons together

Before Autonomous Systems: Machines III

Some steps beyond

- delegating human functions to machines
  - within already existing social structures, organisations, and processes
- creating new functions
  - then, making new social structures, organisations, and processes possible
  - example: steam engines on wheels
- essentially, *changing the world we live in*

Focus on...

Autonomous Software Creatures

Autonomous Robot Toys

Autonomous Vacuum Cleaners

Autonomous Lawnmowers

Autonomous Cars

Autonomous Aircrafts

Autonomous Weapons

Autonomous Soccer Players & Teams

Social Pressure

- activities that might be *delegated* to *artificial systems* grow in *number* and *complexity*
- people & organisations are already experiencing / conceiving "somehow autonomous" systems, and ask for more
- engineers are not yet trained on *general approaches* to build *autonomous systems*
- however, the theory of autonomous systems is at a good stage of development, and technologies are rapidly growing in terms of *availability* and *reliability*

Focus on. . .

Machines & Artificial Systems I

Systems and machines

- we call *systems* what many years ago we simply called *machines*
- complexity has grown
- more and more we understand the many levels at which systems, their components, their mutual relationships can be described
- furthermore, at the right level of abstraction, HW / SW systems are machines in the same acceptation as mechanical machines
- here, we will mostly deal with two non-strictly coherent, but simple notions
  - system as a *primitive notion* (which somehow we all share to a certain extent)
  - system as an *engineer-designed entity* ("draw a line around what you call 'a system'")

Machines & Artificial Systems II

Artificial systems

- here we mostly talk about *artificial systems* in general
- systems either partially or totally designed by humans
  - either directly or indirectly
  - ? systems designed by systems?

  featuring
  - a goal (in the mind of the designer)
  - a function (in the body of the system)

  and implicitly consider the computational part as an essential one
- an artificial system, roughly speaking, is any sort of system which humans put at work by assigning it a function in order to achieve some goal

Which Sorts of Systems? I

Artificial & computational systems

- nowadays, most (if not all) artificial systems have a prominent *computational* part
  - for this and other obvious reasons, here we focus on that sort of systems
- computational machines
  - have both an abstract and a physical part
    - where the physical portions are often abstracted away
  - are (mostly) symbolic
    - can deal with math, logic, data, information, knowledge
  - are general-purpose machines
    - programmable, can be specialised to most purposes

Which Sorts of Systems? II

Artificial systems in context

- most artificial systems *participate* to the activities of individuals, groups, and societies
- even more, nowadays they are mostly essential to all sorts of human activities

Which Sorts of Systems? III

Socio-technical systems (STS)

- socio-technical systems (STS) arise when cognitive and social interaction are mediated by information technology, rather than by the natural world alone [Whitworth, 2006]
- in other words, any system in which the infrastructure enabling and constraining interaction is technological, but the evolution of the system is *driven* by social and cognitive interactions, is a STS
- so, STS are artificial systems where both *humans* and *artificial* components play the role of system components
  - ranging from online reservation systems to social networks
- most of nowadays systems are just STS
  - or, at least, cannot be engineering and successfully put to work without a proper socio-technical perspective in the engineering stage

Which Sorts of Systems? IV

Pervasive systems

- affecting every aspects of our everyday life
- by spreading through the whole environment where we live and act
- we live surrounded by pervasive systems

Situated systems

- the physical nature of artificial components cannot be always be forgot
- as well as the situatedness in time and space
- along with the influence of the surrounding environment
- most of the interesting systems, nowadays, are situated systems, too

What is the Matter with Autonomy? I

Who does what?

- this is no longer an issue
- artificial system are very welcome to do whatever we like

Who takes the decision?

- autonomy is at least as much about *deliberation* as about *action*
- *e.g.*, for artificial weapons, the question is not "who pulls the trigger", but rather "who decides to pull the trigger"
- here, ethical issues become more than relevant

What is the Matter with Autonomy? II

Who is *responsible*?

- what is something goes wrong?
- who is going to take responsibility – under either civil law or criminal law?
- legal issues are here as least as relevant as technical issues

Next in Line. . .

1. Premises
2. On the Notion of Autonomy
3. Intentional Agents

Focus on...

Oxford Dictionary of English (2nd Edition revised 2005)

Etimology

Early 17th cent.: from Greek *autonomia*, from *autonomos* ‘having its own laws’, from *autos* ‘self’ + *nomos* ‘law’.

Dictionary

autonomy

- the right or condition of self-government
- a self-governing country or region
- freedom from external control or influence; independence.
- (in Kantian moral philosophy) the capacity of an agent to act in accordance with objective morality rather than under the influence of desires

Oxford Thesaurus of English (2nd Edition revised 2008)

Thesaurus

autonomy

- self-government, independence, self-rule, home rule, sovereignty, self-determination, freedom, autarchy;
- self-sufficiency, individualism.

Merriam-Webster I

Dictionary

autonomy

1. the quality or state of being self-governing; especially: the right of self-government
2. self-directing freedom and especially moral independence
3. a self-governing state

synonyms accord, free will, choice, self-determination, volition, will

antonyms dependence (also dependance), heteronomy, subjection, unfreedom

Merriam-Webster II

Thesaurus

autonomy

1. the act or power of making one's own choices or decisions: accord, free will, choice, self-determination, volition, will
2. the state of being free from the control or power of another: freedom, independence, independency, liberty, self-determination, self-governance, self-government, sovereignty

Focus on. . .

Autonomy and Biological Systems

Living systems. . .

. . . are the first systems provided of (some level of) autonomy that we have knowledge of

- they *work* as autonomous systems
- they *evolved* to become autonomous

The study of autonomy in biological systems

- the hierarchy of living systems provide examples of many *different* levels of autonomy—from lower to higher levels of autonomy
- the evolutionary view over biological systems potentially sheds light over the *role* of autonomy in living systems

→ the study of living system may help us understanding the many different sorts of autonomy, and their role in (artificial) systems in general

Biology & Evolutionary Biology

Biology

. . . is the study of *living organisms*, or – perhaps more generally – of living systems: their structure, function, growth, origin, evolution, and distribution

Evolutionary biology

. . . studies how evolutionary processes produced diversity of life on Earth; that is, *how biological systems evolved over the ages* [Gould, 2002]. As a result, the view of evolutionary biology over biological systems includes

- not just how they are made, and how they do *work*
- but also – and mainly – how they do evolved towards their current form—the one we can presently observe

Biology vs. Evolutionary Biology

- *evolutionary biology* is indeed a (nowadays constitutive) part of biology
- however, the way evolutionary biologists look over biological systems tends to be less analytical – yet wider – than the one by biologists
- including
  - an overall view of the life on Earth overcoming space and time boundaries
  - the role and mutual influences of all organisms, species, and ecosystems that inhabit (or, have inhabited) our planet

Roughly speaking, evolutionary biology provides us with a *global view over biological systems* at every conceivable level

Evolution & Diversity I

Not just genetic expression

- genes (and their products) are at the core of ours evolutionary model(s)
- however, they alone cannot explain the full range of variation and diversity of living systems
- according to molecular biology, even distantly-related organisms use similar processes for cellular function, development, and metabolism [Rosslenbroich, 2014]
  - bacteria and humans share part of the same metabolism
  - microscopic fungi and humans exhibit a very similar basic cell organisation and functions
  - and so on and so forth

  → most processes are *conserved* during evolution

Evolution & Diversity II

Theory of *facilitation variation* [Kirschner and Gerhart, 2006]

- while new features emerge without forerunners in more ancestral organisms, “the core cellular processes” are conserved throughout evolution
- “a surprisingly small number of genes for humans and complex animal forms reflects the anatomical and physiological complexity that can be achieved by the reuse of genetic products” [Rosslenbroich, 2014]

Evolution & Diversity III

A key question in evolutionary biology

So

- how can we explain the *huge diversity* of life despite its deep and pervasively similar molecular architecture? [Rosslenbroich, 2014]
- how do organisms, species, and life overall *progress* along their evolutionary path?
- and, *do* they actually progress?

Progress in Evolution I

A controversial issue

According to [Rosslenbroich, 2014], the acceptation of the word *progress* conveys a number of diverse meanings—from Darwin to contemporary biologists

1. *change* leads to new (*higher*) organisms
2. which are somehow *improved*
3. progression is *linear*
4. evolution is an *intrinsic force* driving such a process
5. progression has a *goal* / end / culmination / perfection

Nowadays, we just need the first acceptation: however, we need to understand what is *higher*, what is *change*, and *what* is actually progressing along with evolution

Progress in Evolution II

Progress of what?

- increased *potential* for survival?
- increased *efficiency* of some form, like, energy consumption?
- increased *amount of information* in genes?
- increased *differentiation*?
- increased *complexity*?
- increased *emancipation from the environment*?

Nowadays, *complexity* is often used instead of *progress*: this, however, does not offer any explanation—no clear large-scale *patterns* in evolution, here

Autopoiesis I

Autopoiesis is the ability of a complex system of maintaining its own overall coherence, in terms of structure and organisation, through the mutual *interactions* of its components

Autopoiesis II

Living systems as autopoietic systems

[Maturana and Varela, 1980, Varela et al., 1974]

- living systems as autopoietic units capable of sustaining themselves based on an inner network of reactions that generate and regenerate all the system components
- all pertinent processes required have an inner efficient cause
- *structures* – based on a flow of molecules and energy – produce the components that, in turn, continue to maintain the organised bounded structure that gives rise to these components
- self-reference and self-maintenance are core notions here
- coherent and ordered global system behaviour of the system constrains / governs the behaviour of the individual components, while the component behaviour sustains the global order (*circular causality*)

Autopoiesis & Autonomy [Thompson, 2010] I

According to [Maturana and Varela, 1980], autonomous systems

- *acquire* the property of specifying their own rules of behaviour
- do *not* work as transducers or functions for converting input instructions into output products
- are the sources of their own activity, which specify their own domains of interaction

Autopoiesis & Autonomy [Thompson, 2010] II

"In fact, the notion of autopoiesis can be described as a characterisation of the mechanisms which endow living systems with the property of being autonomous; autopoiesis is an explication of the autonomy of the living" [Maturana and Varela, 1980]

Autonomy of a Cell [Thompson, 2010, Rosslenbroich, 2014]

An example of biological autonomy: the cell

- the cell stands out of a molecular soup by *actively creating the boundaries* that
  - set the cell apart from what it is not the cell
  - and simultaneously regulate cell interaction with the environment
- metabolic processes within the cell construct those boundaries, but the metabolic processes themselves are made possible by those boundaries
- thus, the cell emerges as a figure *standing out of a chemical background*
- should this process of self-production be interrupted, the cellular components no longer form a unit, gradually diffusing back into a molecular soup—*death*

Autopoiesis & Autonomy [Thompson, 2010] I

Boundary

- boundary is a central element of autonomy
- it is a constitutive element of the identity of a system
- in a cell, the *membrane* works as a boundary both containing processes/components and regulating the interaction with the environment
- *boundaries* – strict physical ones, not necessarily material – are *essential for an autonomous system*

Autopoiesis & Autonomy [Thompson, 2010] II

Autonomous systems are *closed* [Rosslenbroich, 2014]

- *organisationally* closed in the sense that their organisation is characterised by their internal network processes
- which recursively *depend on each other*, thus constitute the system *as a unit*

Heteronymous vs. Autonomous Systems

[Thompson, 2010, Rosslenbroich, 2014] I

Heteronymous systems

A heteronymous system is one whose organisation is defined by *input-output* information flow and *external* mechanisms of *control*

- traditional computational systems and many network views, for example, are heteronymous
  - they have an input layer and an output layer
  - the inputs are initially assigned by the observer outside the system
  - output performance is evaluated in relation to an externally imposed task

e.g., a Turing Machine typically represents computation by a heteronomous system

Heteronymous vs. Autonomous Systems

[Thompson, 2010, Rosslenbroich, 2014] II

Autonomous systems

An autonomous system is defined by its *endogenous*, self-organising, and self-controlling *dynamics*, and determines the domain in which it operates

- it has *input* and *output*—which, alone, do not determine the system
- it is the internal *self-production* process that *controls* and *regulates* the system's *interaction* with the outside *environment*

Autonomy and Environment I

Autonomy is not autarky

- living systems are *not independent* of their *environment*
- the *interchange* occurs though the *physical boundary*

Autonomy and Environment II

Plants vs. animals

- plants exhibit a predominantly *open* relation to their environment
- instead, animals have a more *closed* form of organisation
  - the exchange surfaces for metabolism are turned to the inside
  - special internal organs and internal cavities appear
  - exchange surfaces on the outside are reduced
- the loss of a direct environmental relation corresponds to a gain in degrees of freedom
- *stimulus-response* relationships in animals tend to be less tightly connected
- in animals, signals can internally be enforced, compared to other signals, and memorised
- thus, not a rigid, but rather a flexible relation between organism and environment emerges when "moving up" from plants to animals

Robustness

Stability in front of change

- many structures and functions – as well as proteins and genes – have certain stability in the face of *environmental variations* and *genetic changes*
- they are resistant, robust, to *perturbations*, producing relatively invariant outputs [Kitano, 2002]

Robustness in Living Systems

Robustness

- is understood as a property that allows a system to *maintain its functions* against internal and external perturbations and uncertainties
- encompasses a broad range of traits: from macroscopic, visible traits, to molecular traits, such as the expression level of a gene, or, the three-dimensional conformation of a protein
- is widely recognised as an *inherent property* of all biological systems

Autonomy & Robustness

- autonomy and robustness somehow overlap, but they are not the same
- robustness may be seen as a *pre-requisite* for autonomy: for instance, self-maintenance requires robustness
- or, robustness is a *part of* autonomy, as it maintains the identity, structure, and organisation of a living system against well-separated surroundings

Principles & Strategies for Robustness [Kitano, 2002]

redundancy of components to protect *against failure* of a specific component by providing for alternative ways to carry out the function the component performs

feedback circuits to *monitor* a system function so as to *regulate* it

modularity as the *encapsulation* of functions, for robustness and evolvability

layering in hierarchical systems to enhance *control* and robustness

Homeostasis

Homeostasis is the ability of a system to regulate its internal conditions to keep some or several functions stable

e.g., properties such as temperature or blood composition in animals

- separating internal and external environments
- where *internal environment* is kept relatively stable with respect to *external perturbations*

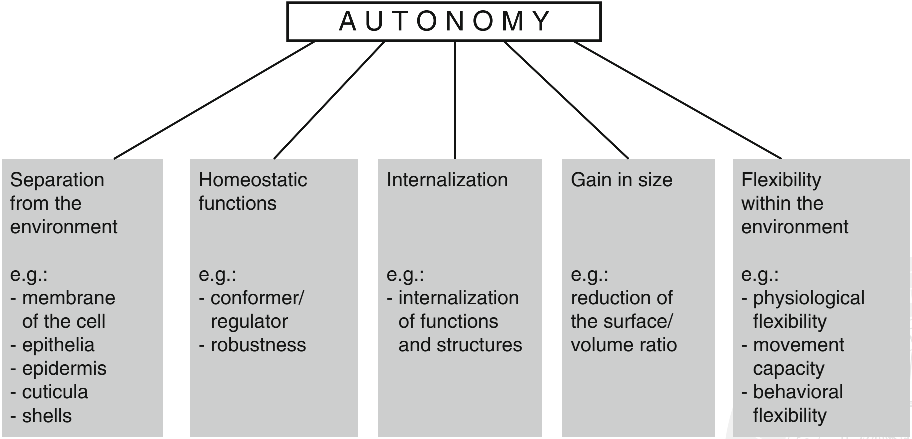
Time Autonomy

- living entities establish *their own* cycles in *time*
- e.g., metabolism, rest-activity cycle, development, reproduction
- involving all biochemical, cellular, and organic processes
- from reaction rates, frequencies are endogenous, lead to *autonomous cycles*, which only later synchronise with external cycles

Evolutions through Increasing Autonomy

Increasing autonomy [Rosslenbroich, 2014]

- which features are able to contribute to *changes* in autonomy of an individual organism?
- how can autonomy be *defined* accordingly in a more formal way?



Resources to change autonomy—from [Rosslenbroich, 2014]

A Definition for Autonomy in Living Systems

General autonomy [Rosslenbroich, 2014]

Living systems are *autonomous* in the sense that they *maintain themselves in form and function* within time and achieve a *self-determined flexibility*

1. they generate, maintain, and regulate an *inner network* of interdependent, energy-consuming processes, which in turn generate and maintain the system
2. they establish a *boundary* and actively regulate their *interaction* and exchange with the *environment*
3. they specify their *own rules of behaviour* and react to external stimuli in a *self-determined* way, according to their internal disposition and condition
4. they establish an *interdependence* between the system and its parts within the organism, which includes a *differentiation* in subsystems
5. they establish a *time autonomy*
6. they maintain a phenotypic stability (*robustness*) in the face of diverse perturbations arising from environmental changes, internal variability, and genetic variations

Autonomy in Evolution

- autonomy is an essential trait of living systems
- many authors observe that evolution progresses through biological systems with *diverse degrees of autonomy*
- many additional functions resulting from evolutionary processes tends to improve autonomy
- how can autonomy help in understanding evolution?
- is it a *pattern of evolution*?

Increasing Autonomy [Rosslenbroich, 2014] I

- increasing autonomy is defined as an *evolutionary shift* in the *system-environment relationship*, such that

  interactive autonomy the direct influences of the environment on the respective individual systems are gradually reduced

  constitutive autonomy stability and flexibility of self-referential, intrinsic functions within the systems are generated and enhanced

- with respect to the environment
  - *autonomy* of living systems is *relative*
  - while retaining numerous interconnections with and dependencies on the external environment
  - thus, organisms can undergo relative *emancipation from environmental fluctuations*, gaining self-determination and flexibility of behaviour

Increasing Autonomy [Rosslenbroich, 2014] II

- a set of resources can be involved to change autonomous capacities
  1. changes in *spatial separation* from the environment
  2. changes in *homeostatic* capacities and robustness
  3. *internalisation* of structures or functions
  4. increase in body *size*
  5. changes in the *flexibility* within the environment, including behavioural flexibility

Overall: Autonomy as a Driver for Evolution

- autonomy is an essential trait of living systems
- biological systems evolve towards increasing degrees of autonomy
- autonomy is an evolutionary pattern

Focus on. . .

Robots Playing Music: Which Autonomy?

Internet Encyclopedia of Philosophy I

Many acceptations of autonomy

general an individual's capacity for self-determination or self-governance

folk inchoate desire for freedom in some area of one's life

personal the capacity to decide for oneself and pursue a course of action in one's life

moral the capacity to deliberate and to give oneself the moral law, rather than merely heeding the injunctions of others

political the property of having one's decisions respected, honored, and heeded within a political context

Internet Encyclopedia of Philosophy II

Individual autonomy

- after Kant, autonomy is an essential trait of the individual, and strictly related with its morality, represented by some high-level ethical principles
- then, with the relation between its inner self and its individual actions
- that is, mind and behaviour

Internet Encyclopedia of Philosophy III

Independence from oneself

- a more demanding notion of autonomy requires not only self-determination, but also independence from oneself
- this conception is connected with notions of freedom and choice, and (maybe) non-determinism
- and requires the ability of reasoning on (and possibly changing) not just one own course of actions, but one own goals

Focus on. . .

Unmanned Systems Integrated Roadmap FY 2011-2036 I

Automatic vs. autonomous

automatic systems are fully pre-programmed and act repeatedly and independently of external influence or control. An automatic system can be described as self-steering or self-regulating and is able to follow an externally given path while compensating for small deviations caused by external disturbances. However, the automatic system is not able to define the path according to some given goal or to choose the goal dictating its path.

autonomous systems are self-directed toward a *goal* in that they do not require outside control, but rather are governed by laws and strategies that direct their behavior. Initially, these control algorithms are created and tested by teams of human operators and software developers. However, if *machine learning* is utilized, autonomous systems can develop modified strategies for themselves by which they select their behavior. An autonomous system is self-directed by choosing the behavior it follows to reach a human-directed goal.

Unmanned Systems Integrated Roadmap FY 2011-2036 II

Four levels of autonomy for unmanned systems [Edwards, 2013]

Various levels of autonomy in any system guide how much and how often humans need to interact or intervene with the autonomous system:

- human operated
- human delegated
- human supervised
- fully autonomous

Unmanned Systems Integrated Roadmap FY 2011-2036 III

Level	Name	Description
1	Human Operated	A human operator makes all decisions. The system has no autonomous control of its environment although it may have information-only responses to sensed data.
2	Human Delegated	The vehicle can perform many functions independently of human control when delegated to do so. This level encompasses automatic controls, engine controls, and other low-level automation that must be activated or deactivated by human input and must act in mutual exclusion of human operation.
3	Human Supervised	The system can perform a wide variety of activities when given top-level permissions or direction by a human. Both the human and the system can initiate behaviors based on sensed data, but the system can do so only if within the scope of its currently directed tasks.
4	Fully Autonomous	The system receives goals from humans and translates them into tasks to be performed without human interaction. A human could still enter the loop in an emergency or change the goals, although in practice there may be significant time delays before human intervention occurs.

Unmanned Systems Integrated Roadmap FY 2011-2036 IV

Autonomy & unpredictability

- the special feature of an autonomous system is its ability to be goal-directed in *unpredictable situations*.
- this ability is a significant improvement in capability compared to the capabilities of automatic systems.
- an autonomous system is able to *make a decision* based on a set of rules and/or limitations.
- it is able to determine *what information is important* in making a decision.
- it is capable of a *higher level of performance* compared to the performance of a system operating in a predetermined manner.

Autonomy in AWS I

Who takes the decision?

- autonomy is at least about *deliberation* as much as about *action*
- *e.g.*, for artificial weapons, the question is not just

  who pulls the trigger?

  but also / rather

  who decides to pull the trigger?

  and

  based on what evidence?

Autonomy in AWS II

Autonomy & responsibility

- *responsibility* is both an ethical and a legal issue
- autonomy of AWS changes the overall picture
- *multi-level autonomy* has the potential to make things even much more complicated [Sartor and Omicini, 2016]

Focus on. . .

Autonomy as a Relational Concept [Castelfranchi, 1995] I

Autonomy as a social concept

- an agent is autonomous mostly in relation to other agents
- autonomy has no meaning for an agent in isolation

Autonomy from environment

- the Descartes' problem: (human, agent) behaviour is affected by the environment, but is not depending on the environment
- situatedness, reactiveness, adaptiveness do not imply lack of autonomy

Autonomous Goals [Castelfranchi, 1995] I

Agency

- agents as teleonomic / teleologic, *goal-driven* entities
- that is, whose *behaviour* is *not casual* under any acceptation of the term

$\rightarrow$ it might be non-deterministic, never casual

Autonomous Goals [Castelfranchi, 1995] II

Agents & goals [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*

Autonomous Goals [Castelfranchi, 1995] III

Executive vs. motivational autonomy

executive autonomy — given a goal, the agent is autonomous in achieving it by itself

motivational autonomy the agent's goals are somehow self-generated, not externally imposed

Autonomy & autonomous goals

- autonomy requires *autonomous goals*
- executive autonomy is not enough for real autonomy

Goal-autonomous agent

A *goal-autonomous agent* is an agent endowed with its own *goals*

Autonomous Goals [Castelfranchi, 1995] IV

An agent is *fully socially autonomous* if

1. it has its own goals: *endogenous*, not derived from other agents' will
2. it is able to *make decisions* concerning *multiple conflicting goals* (being them its own goals or also goals adopted from outside)
3. it adopts *goals from outside*, from other agents; it is liable to *influencing*
4. it adopts other agents' goals as a consequence of a *choice* among them and other goals
5. it adopts other agents' goals only if it sees the adoption as a way of enabling itself to achieve some of its own goals (i.e., the autonomous agent is a *self-interested* agent)
    - it is not possible to directly modify the agent's goals from outside: any modification of its goals must be achieved by modifying its *beliefs*
    - thus, the control over beliefs becomes a filter, an additional control over the adoption of goals

Autonomous Goals [Castelfranchi, 1995] V

6. it is impossible to change automatically the beliefs of an agent
   - the adoption of a belief is a special "decision" that the agent takes on the basis of many criteria
   - this protects its *cognitive autonomy*

Focus on. . .

Evolution of Programming Languages: The Picture

	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)

[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

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

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

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
- agents are autonomous entities

Focus on. . .

Autonomy as the Foundation of the Definition of Agent

Lex Parsimoniae: Autonomy

- autonomy as the only fundamental and defining 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*

Agents are Reactive I

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

Agents are Reactive II

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
- however, this issue will be even clearer when facing the issue of artifacts and environment design

(Autonomous) Agents Change the World

Action, change & environment

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

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

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

Autonomous Agents are Social

From autonomy to society

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

[Graesser, 1996, Odell, 2002]

Autonomous agents live in a MAS

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

Autonomous Agents are Interactive

Interactivity 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* 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

Agents as Autonomous Components

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*

Next in Line...

Intelligent Agents I

According to a classical definition, an intelligent agent is a computational system capable of autonomous action and perception in some environment

Reminder: 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

Intelligent Agents II

Question: what about the other notions of autonomy?

- autonomy with respect to other agents – social autonomy
- autonomy with respect to environment – interactive autonomy
- autonomy with respect to humans – artificial autonomy
- autonomy with respect to oneself – moral autonomy
- ...

Question: what is intelligence to autonomy?

- any sort of intelligence?
- which intelligence for which autonomy?
- which intelligent architecture for which autonomy?
- ...

Focus on. . .

Intentional Systems

- an idea is to refer to human attitudes as *intentional notions*
- when explaining human activity, it is often useful to make statements such as the following:
  - Seb got rain tires because he believed it was going to rain
  - Kimi is working hard because he wants to win world championship again
- these statements can be read in terms of *folk psychology*, by which human behaviour can be explained and can be *predicted* through the attribution of mental attitudes, such as believing and wanting (as in the above examples), hoping, fearing, and so on.

The Intentional Stance

- the philosopher – cognitive scientist – Daniel Dennett coined the term intentional system to describe entities *'whose behaviour can be predicted by the method of attributing to it belief, desires and rational acumen'* [Dennett, 1971]
- Dennett identifies several grades of intentional systems:
  1. a first-order intentional system has beliefs, desires, etc.
     - Seb believes P
  2. a second-order intentional system has beliefs, desires, etc. about beliefs, desires, etc. both its own and of others
     - Seb believes that Kimi believes P
  3. a third-order intentional system is then something like
     - Seb believes that Kimi believes that Seb believes P
  4. ...

The Intentional Stance in Computing

What entities can be described in terms of *intentional stance*?

- human beings are prone to provide an intentional stance to almost anything
  - sacrifices for ingratiating gods benevolence
  - animism
  - ...

Ascribing mental qualities to machines [McCarthy, 1979]

Ascribing mental qualities like beliefs, intentions and wants to a machine is sometimes correct if done conservatively and is sometimes necessary to express what is known about its state [...] it is useful when the ascription helps us understand the structure of the machine, its past or future behaviour, or how to repair or improve it

Focus on. . .

Agents as Intentional Systems

Strong notion of agency

- early agent theorists start from a (strong) notion of agents as intentional systems
- agents were explained in terms of mental attitudes, or mental states
- in their social abilities, agents simplest consistent description implied the intentional stance
- agents contain an explicitly-represented – *symbolic* – model of the world (written somewhere in the working memory)
- agents make decision on what action to take in order to achieve their goals via *symbolic* reasoning

Which Domains for Intention Systems? I

Mental states are a worth *abstraction* for developing agents to effectively act in a class of application *domains* characterised by various practical limitations and *requirements* [Rao and Georgeff, 1995]

Which Domains for Intention Systems? II

- at any instant of time there are many different ways in which an environment may evolve—the environment is not deterministic
- at any instant of time there are many actions or procedures the agent may execute—the agent is not deterministic, too
- at any instant of time the agent may want to achieve several objectives
- the actions or procedures that (best) achieve the various objectives are dependent on the state of the environment—i.e., on the particular situation, context
- the environment can only be sensed locally
- the rate at which computations and actions can be carried out is within *reasonable bounds* to the rate at which the environment evolves

Goal-Oriented & Goal-Directed Systems

- there are two main families of architectures for agents with mental states

  teleo-reactive / goal-oriented agents
  : are based on their own design model and internal control mechanism. The *goal* is not explicitly represented within the internal state, instead it is an 'end state' for agents internal state machine

  deliberative / goal-directed agents
  : are based on symbolic reasoning about goals, which are explicitly represented and processed aside the control loop

Modelling Agents with Mental States I

Modelling agents based on mental states. . .

- eases the development of agents exhibiting complex behaviour
- provides us with a familiar, non-technical way of understanding and explaining agents
- allows the developer to build MAS by adopting the perspective of a cognitive entity engaged in complex tasks—e.g., what would *I* do in the same situation?
- simplifies the construction, maintenance, and verification of agent-based applications
- is useful when the agent has to comunicate and interact with users or other system entities

Modelling Agents with Mental States II

The intentional stance [Dennett, 2007]

The intentional stance is the strategy of interpreting the behaviour of an entity (person, animal, artifact, whatever) by treating it as if it were a rational agent who governed its 'choice' of 'action' by a 'consideration' of its 'beliefs' and 'desires'.

The scare-quotes around all these terms draw attention to the fact that some of their standard connotations may be set aside in the interests of exploiting their central features: their role in practical reasoning, and hence in the prediction of the behaviour of practical reasoners.

Modelling Agents with Mental States III

Agents with mental states

- agents governing their behaviour based on *internal states* that mimic cognitive (human) *mental states*

  epistemic states representing agents *knowledge*—their knowledge on the world
  - i.e., percepts, beliefs

  motivational states representing agents *objectives*—what they aim to achieve
  - i.e., goals, desires
- the process of selecting one action to execute among the many available based on the actual mental states is called *practical reasoning*
  - i.e., $action(next(i, perception(e))$

Focus on. . .

Practical vs. Epistemic Reasoning

practical reasoning is reasoning directed towards actions—the process of figuring out what to do in order to achieve what is desired

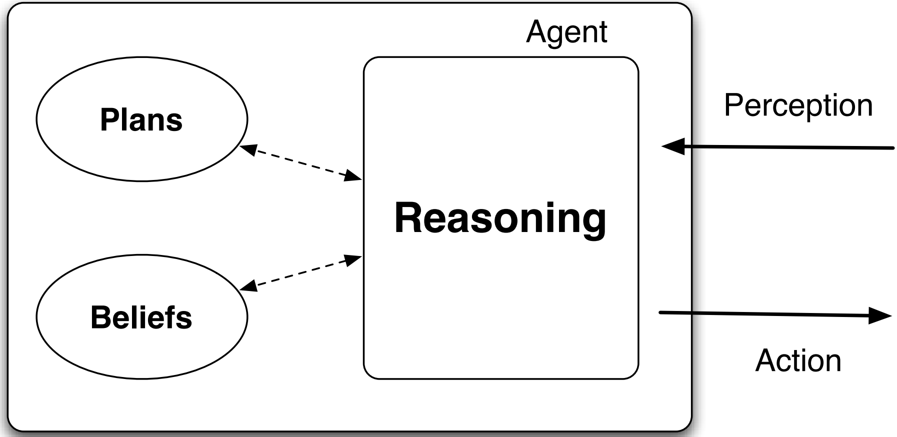
> [Bratman, 1987]
>
> Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes.

epistemic reasoning is reasoning directed towards knowledge—the process of updating information, replacing old information (no longer consistent with the world state) with new information

Practical Reasoning

- practical reasoning consists of two main cognitive activities
  - **deliberation** when the agent makes decision on what state of affairs the agent desire to achieve
  - **means-ends reasoning** when the agent makes decisions on how to achieve these state of affairs
- the outcome of the deliberation phase are the *intentions*
  - what agent desires to achieve, or what he desires to do
- the outcome of the means-ends reasoning phase is the selection a given course of actions
  - the workflow of the actions the agent intends to adopt in order to achieve its own goals expressed as intentions

Basic Architecture of a Mentalistic Agent



The Role of Intentions in Practical Reasoning I

1. intentions represent a problem to solve for the agent who need to determine how to achieve them
   - if I have an intention to ϕ, you would expect me to *devote resources* to deciding how to bring about ϕ.
2. intentions provide a *filter* for adopting other intentions, which must *not conflict*
   - if I have an intention to ϕ, you would not expect me to adopt an intention ψ such that ϕ and ψ are mutually exclusive
3. intentions tend to be *stable*: agents track the success of their intentions, and are inclined to try again if their attempts fail
   - if an agent's first attempt to achieve ϕ fails, then all other things being equal, it will try an alternative plan to achieve ϕ
4. agents believe their intentions are *possible*
   - that is, they believe that there is at least some way that the intentions could be brought about

The Role of Intentions in Practical Reasoning II

5. agents do *not believe* they will *not bring about* their intentions.
   - it would not be rational for me to adopt an intention to ϕ if I believed ϕ was not possible.
6. under certain circumstances, agents believe they will *bring about their intentions*
   - it would not normally be rational of me to believe that I would bring my intentions about; intentions can fail
   - moreover, it does not make sense that if I believe ϕ is inevitable that I would adopt it as an intention
7. agents need not intend all the expected side effects of their intentions
   - if I believe $\phi \rightarrow \psi$ and I intend that ϕ, I do not necessarily intend ψ also

   $\rightarrow$ intentions are not closed under implication
   - this last problem is known as the *side effect* or *package deal* problem: I may believe that going to the dentist involves pain, and I may also intend to go to the dentist—but this does not imply in any way that I intend to suffer pain

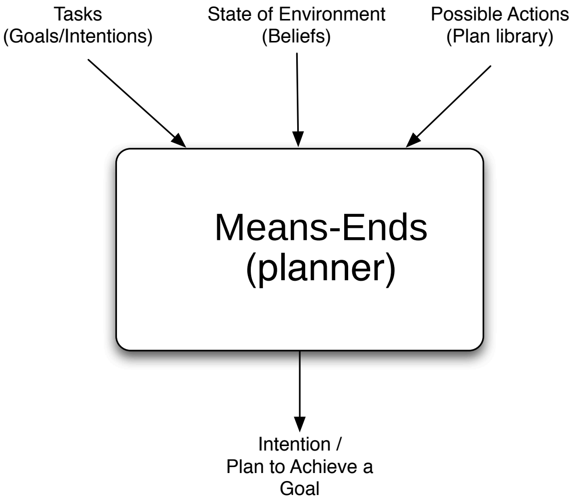
Intentions vs. Desires

- the adoption of an intention follows the rise of a given *desire*
  - i.e., it follows the adoption of a given goal
- desires and intentions are different concepts
  - "My desire to play basketball this afternoon is merely a potential influencer of my conduct this afternoon. It must live with my other relevant desires [...] before it is settled what I will do".
  - "In contrast, once I intend to play basketball this afternoon, the matter is settled: I normally need not continue to weigh the pros and cons. When the afternoon arrives, I will normally just proceed to execute my intentions." [Bratman, 1990]

Means-Ends Reasoning I

- the basic idea is to provide agents with three sorts of representations
  - representation of goal / intention to achieve
  - representation of actions / plans – in repertoire
  - representation of the environment
- given the environmental conditions, means-ends reasoning aims at *devising out a plan* that could possibly *achieve the adopted goal* / intention
- the selected intention is an emergent property, reified at runtime by selecting a given plan for achieving a given goal

Means-Ends Reasoning II



Focus on. . .

Implementing a Practical Reasoning Agent: Issues I

Problem

Agents have *bounded resources*—what is called bounded rationality

- deliberation and means-ends processes are not for free: they have *computational costs*
- the time taken to reason and the time taken to act are potentially *unbounded*

→ this harms agent *fitness*—that is, the reactivity and the promptness that is essential for the agent to survive

Implementing a Practical Reasoning Agent: Issues II

- if the agent
  - starts deliberating at t_0
  - begins means-ends at t_1
  - begins executing a plan at t_2
  - ends executing a plan at t_3
- then
  - *time for deliberation* is

$$t_{deliberation} = t_1 - t_0$$

  - *time for means-ends reasoning* is

$$t_{meansend} = t_2 - t_1$$

Implementing a Practical Reasoning Agent: Issues III

- *agents environments* are supposed to be highly dynamic
  - many concurrent changes may occur during agent decision-making as well as during the execution of plans
- the deliberated intention is surely worth to be pursued at the precise time when it the deliberation process starts—so, at t_0
- at time t_1, the agent *selects* a goal/intention that would have been optimal if it had been *achieved* at t_0
- the agent runs the risk that the intention selected is no longer optimal – or no longer achievable – by the time the agent has committed to it

Implementing a Practical Reasoning Agent: Issues IV

So, this agent will exhibit an overall optimal behaviour in the following circumstances / under the following conditions:

1. when deliberation and means-ends reasoning take a vanishingly-small amount of time
2. when the world is guaranteed to remain (essentially) static while the agent is deliberating and performing means-ends reasoning, so that the assumptions upon which the choice of intention to achieve and plan to achieve the intention remain valid until the agent has completed both deliberation and means-ends reasoning
3. when an intention that is optimal when achieved at t_0 – the time at which the world is observed – is guaranteed to remain optimal until t_2—the time at which the agent has found a course of action to achieve the intention

The BDI Framework I

According to [Dasgupta and Ghose, 2011]

- one of the most popular and successful framework for agent technology is defined by Rao and Georgeff [Rao and Georgeff, 1992]
- there, the notions of *belief*, *desire*, and *intention* are the core ones
- hence, agents in this framework are typically referred to as *BDI agents*

The BDI Framework II

beliefs represent at any time the agent's current *knowledge about the world*, including

- information about the current state of the environment inferred from perception devices
- messages from other agents
- internal information

desires represent a *state of the world* the agent is trying *to achieve*

intentions are the chosen means to achieve the agent's desires, and are generally implemented as plans and post-conditions

- as in general it may have *multiple desires*, an agent can have *a number of intentions active at any one time*
- these intentions may be thought of as *running concurrently*, with *one chosen intention active at any one time*

The BDI Framework III

Besides these components, the BDI model includes

plan library — namely, a set of "recipes" representing the *procedural knowledge* of the agent

event queue — where

- *events* — either perceived from the environment or generated by the agent itself to notify an update of its belief base
- *internal subgoals* — generated by the agent itself while trying to achieve a desire

are stored.

The BDI Framework IV

Plans & plan library

- usually, BDI-style agents do no adopt first principles planning at all
- all plans must be *generated* by the agent programmer *at design time*, which are then *selected* for execution at run time
- pre-programmed plans are collected in the plan library
- the planning done by agents consists entirely of context-sensitive subgoal expansion, which is deferred until a subgoal is selected for execution

The BDI Abstract Architecture

Accordingly, the *abstract architecture* proposed by [Rao and Georgeff, 1992] comprise three dynamic and global structures representing agent beliefs, desires, and intentions (BDI), along with an input queue of events

- *update* (write) and *query* (read) operations are possible upon the three structures
  - update operation are subject to compatibility requirements
  - formalised constraints hold upon the mental attitudes
- the events that the system is able to recognise could be either external – i.e., coming from the environment – or internal ones—i.e., coming from some reflexive action
  - events are assumed to be *atomic*, and can be recognised after they have occurred

Implementing a BDI Agent

1. the agent initialises the internal states
2. the agent enters the main loop
3. the option generator reads the event queue, and returns a list of options
4. the deliberator selects a subset of options to be adopted, and adds these to the intention structure
5. the intentions to be adopted are filtered from the selected ones
6. if there is an intention to perform an atomic action at this point in time the agent executes it
7. any external events that have occurred during the interpreter cycle are then added to the event queue (the same for internal events)
8. the agent modifies the intention and the desire structures by dropping successful ones
9. finally, impossible desires and intentions are dropped, too

How does a BDI Agent Deliberate?

Problem

How can we made reasoning procedures of deliberation and option generation sufficiently fast to satisfy the real time demands placed upon the cognitive system?

- deliberation can be decomposed in two phases:
  - option generation — understand what are the available alternatives
  - deliberation — choose (and filter) between the adoptable goals/intentions
- chosen options are then intentions, so the agents commit to the selected ones—and executes them

Refining Deliberation Function I

option generation — the agent generates a set of possible alternatives; represents option generation via a function, options, which takes agent's current beliefs and current intentions, and from them determines a set of options (i.e., desires)

deliberation — the agent chooses between *competing* alternatives, and commits to the intention to achieving them; in order to select between competing options, an agent uses a *filter* function

Refining Deliberation Function II

Notes

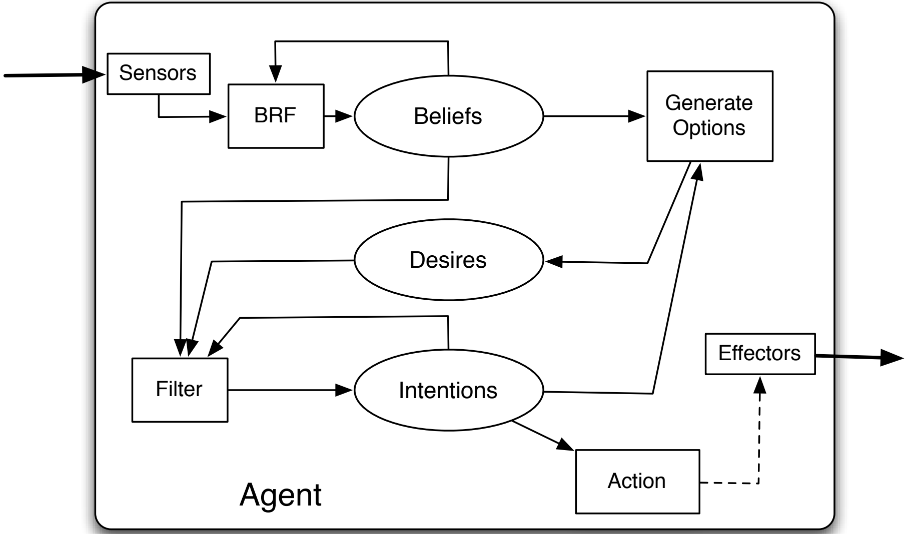
- the strategy for deliberating between goals typically is in the hands of the agent developer
- most BDI programming platforms provide mechanisms to describe under which conditions some goal should inhibit the others (*goal formulae*)
- typically, such goal formulae are first-order logic predicates indicating contexts and trigger conditions
- *game theory* can enter the picture, here: i.e., maximising expected utilities

Structure of BDI Systems

BDI architectures are based on the following constructs

- a set of beliefs
- a set of desires (or goals)
- a set of intentions
  - or better, a subset of the goals with an associated stack of plans for achieving them; these are the intended actions
- a set of internal events
  - elicited by a belief change (i.e., updates, addition, deletion) or by goal events (i.e. a goal achievement, or a new goal adoption)
- a set of external events
  - perceptive events coming form the interaction with external entities (i.e. message arrival, signals, etc.)
- a plan library (repertoire of actions) as a further (static) component

Basic Architecture of a BDI Agent [Wooldridge, 2002]



Post-Declarative Systems

It was said that this approach leads to a kind of *post-declarative* programming

- in procedural programming, we say exactly what a system should do
- in declarative programming, we state something that we want to achieve, give the system general info about the relationships between objects, and let a built-in control mechanism (e.g., goal-directed theorem proving) figure out what to do
- with intentional agents, we give a very abstract specification of the system, and let the control mechanism figure out what to do, knowing that it will act in accordance with the built-in theory of agency

Actually, the BDI framework combines in an excellent way both (post)declarative structure and procedural knowledge in terms of plans

Beliefs

Beliefs

Agents knowledge is structured in beliefs about the current state of the world

- they are informational units, typically implemented as ground sets of literals, possibly with no disjunctions or implications
- they should reflect only the information which is currently held (i.e. situated)
- they are expected to change in the future, i.e., as well as the environment changes

Plans

Plans

Plans represent the means the agent has to change the world, and to bring it closer to his desires

- they are language constructs, typically implemented in the form of procedural structures
- plans have a 'body', describing the workflow of activities (actions) that have to be executed for plan execution to be successful
- the conditions under which a plan can be chosen as an option are specified in an *invocation condition* (triggering event) and a *pre- or context- condition* (situation that must hold for the plan to be executable)

Intentions

Intentions

Intentions are emergent properties reified at runtime by selecting a given plan for achieving a given goal

- represented 'on-line' using a run-time stack of hierarchically plans related to the ongoing adopted goals
- similarly to how Prolog interpreter handle clauses
- multiple intention stacks can coexist, either running in parallel, suspended until some condition occurs, or ordered for execution in some way

BDI Viewpoints I

There are three main viewpoints over the BDI Model [Mazal et al., 2008]:

philosophical based on the work of philosopher Bratman [Bratman, 1987], using uses terms of folk psychology to view humans as planning agents: the main concepts in his work are *beliefs* (what an agent knows about the world), *desires* (what the agent wants, can be contradictory) and *intentions* (desires that the agent has decided to reach, cannot be contradictory)

logical mainly Rao and Georgeff's BDI CTL [Rao and Georgeff, 1998] – multimodal logics with possible world semantics –, providing beliefs, goals (desires), and intentions with a precise logical semantics

BDI Viewpoints II

implementation there are a huge number of systems and technologies that are said to conform to the BDI model—between the BDI CTL logics (very expressive) and the implementing systems. which then treat the main modalities rather as data structures, and mostly focus on plans

BDI Agents Programming Platforms

Jason (Brasil) `http://jason.sourceforge.net/`
Agent platform and language for BDI agents based on AgentSpeak(L)

JADEX (Germany) `http://www.activecomponents.org/`
Agent platform for BDI and Goal-Directed Agents

2APL (Netherlands) `http://www.cs.uu.nl/2apl/`
Agent platform providing programming constructs to implement cognitive agents based on the BDI architecture

3APL (Netherlands) `http://www.cs.uu.nl/3apl/`
A programming language for implementing cognitive agents

PRACTIONIST (Italy) `http://practionist.eng.it/`
Framework built on the Bratman's theory of practical reasoning to support the development of BDI agents

ASTRA `http://astralanguage.com/`
A distributed / concurrent programming language based on agent-oriented programming

Part II

System Autonomy & Self-Organisation

Next in Line...

Complex Systems

> *. . . by a complex system I mean one made up of a large number of parts that interact in a non simple way* [Simon, 1962]

Which "parts" for complex systems?

- is autonomy of "parts" a necessary precondition?
- is it also sufficient?

Which kind of systems are we looking for?

- what is *autonomy* for a system as a whole?
- where could we find significant examples?

Nature-inspired Models

Complex natural systems

- such as physical, chemical, biochemical, biological, social systems
- natural system exhibit *features*
  - such as distribution, opennes, situation, fault tolerance, robustness, adaptiveness, ...
- which we would like to understand, capture, then bring to *computational* systems

Nature-Inspired Computing (NIC)

- for instance, NIC [Liu and Tsui, 2006] summarises decades of research activities
- putting emphasis on
  - autonomy of components
  - self-organisation of systems

Focus on...

4 Autonomy in Complex Artificial Systems
- Self-Organisation
- Self-Organisation and Emergence in Natural Systems
- Self-Organisation and Emergence in Artificial Systems
- Self-Organising Systems & Autonomy
- Multi-Agent Systems vs. Self-Organising Systems

5 Coordination for Self-Organisation & System Autonomy
- Coordination Models
- LINDA & Tuple-based Coordination
- Nature-inspired Coordination
- Tuple-based models for Nature-inspired Coordination
- Coordination Technologies

Intuitive Idea of Self-Organisation

- self-organisation generally refers to the internal process leading to an increasing level of organisation
- *organisation* stands for relations between parts in term of structure and interactions
- *self* means that the driving force must be internal, specifically, distributed among components

History of Self-Organisation

- the idea of the spontaneous creation of organisation can be traced back to René Descartes
- according to the literature, the first occurrence of the term *self-organisation* is due to a 1947 paper by W. Ross Ashby [Ashby, 1947]
- Ashby defined a system to be self-organising if it changed its own organisation, rather being changed from an external entity

Elements of Self-Organisation

increasing order — due to the increasing organisation

autonomy — interaction with external world is allowed as long as the control is not delegated

adaptive — suitably responds to external changes

dynamic — it is a process not a final state

Self-Organisation in Sciences

- initially ignored, the concept of self-organisation is present in almost every science of complexity, including
  - physics
  - chemistry
  - biology and ecology
  - economics
  - artificial intelligence
  - computer science

History of Emergence

- emergence is generally referred as the phenomenon involving global behaviours arising from local components interactions
- although the origin of the term emergence can be traced back to Greeks, the modern meaning is due to the English philosopher G.H. Lewes (1875)
- with respect to chemical reactions, Lewes distinguished between *resultants* and *emergents*
  - resultants are characterised only by their components, i.e. they are reducible
  - conversely, emergents cannot be described in terms of their components

Definition of Emergence

- we adopt the definition of emergence provided in [Goldstein, 1999]

  Emergence [..] refers to the arising of novel and coherent structures, patterns, and properties during the process of self-organisation in complex systems. Emergent phenomena are conceptualised as occurring on the macro level, in contrast to the micro-level components and processes out of which they arise.

Emergence vs. Holism

- emergence is often, and imprecisely, explained resorting to *holism*
- holism is a theory summarisable by the sentence *the whole is more than the sum of the parts*
- while it is true that an emergent pattern cannot be reduced to the behaviour of the individual components, emergence is a more comprehensive concept

Properties of Emergent Phenomena

novelty — unpredictability from low-level components

coherence — a sense of identity maintained over time

macro-level — emergence happens at an higher-level w.r.t. to components

dynamism — arise over time, not pre-given

ostensive — recognised by its manifestation

Requirements for Emergency

Emergence can be exhibited by systems meeting the following requirements

non-linearity — interactions should be non-linear and are typically represented as feedback-loops

self-organisation — the ability to self-regulate and adapt the behaviour

beyond equilibrium — non interested in a final state but on system dynamics

attractors — dynamically stable working state

Definition of Self-Organisation

Widespread definition of self-organisation by [Camazine et al., 2001]

Self-organisation is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern.

- it is evident that the authors conceive self-organisation as the source of emergence
- this tendency of combining emergence and self-organisation is quite common in biological sciences
- in the literature there is plenty of misleading definitions of self-organisation and emergence [De Wolf and Holvoet, 2005]

Focus on...

Natural Systems

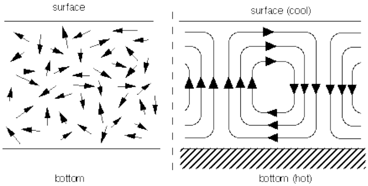
- natural systems can be broadly thought as [Di Marzo Serugendo et al., 2011]
  - physical systems
  - biological systems
  - social systems

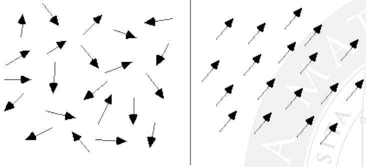
Physics and Chemistry

- theory of self-organisation were originally developed within physics and chemistry
- most typical features included
  - when the system reaches a *critical threshold*, an immediate change occurs
  - self-organisation can be observed *globally*

Self-Organisation of Matter

- self-organisation of matter happens in several fashion
- in magnetisation, spins spontaneously align themselves in order to repel each other, producing and overall strong field
- Bénard cells is a phenomena of convection where molecules arrange themselves in regular patterns because of the temperature gradient



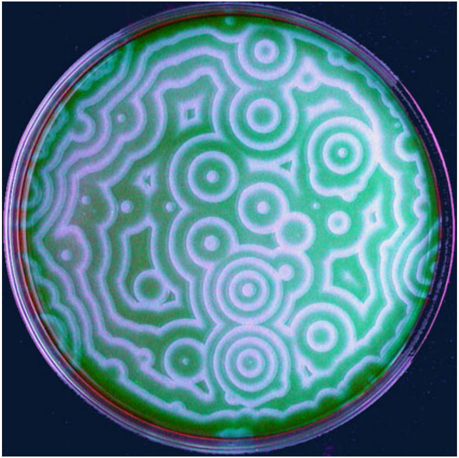


The left hand side picture displays Bénard cells.
The right hand side picture displays magnetisation.

Belousov-Zhabotinsky Reaction I

- discovered by Belousov in the 1950s and later refined by Zhabontinsky, BZ reactions are a typical example of far-from-equilibrium system
- mixing chemical reactants in proper quantities, the solution colour or patterns tend to oscillate
- these solutions are referred as *chemical oscillators*
- there have been discovered several reactions behaving as oscillators

Belousov-Zhabotinsky Reaction II



A snapshot of the Belousov-Zhabotinsky reaction.

Living Organisms

- self-organisation is a common phenomenon in subsystems of living organisms
- an important field in biological research is the determination of invariants in the evolution of living organisms
  - in particular the spontaneous appearance of order in living complex systems due to self-organisation
- in biological research, self-organisation essentially means the global emergence of a particular behaviour or feature that cannot be reduced to the properties of individual system's components—such as molecules and cells

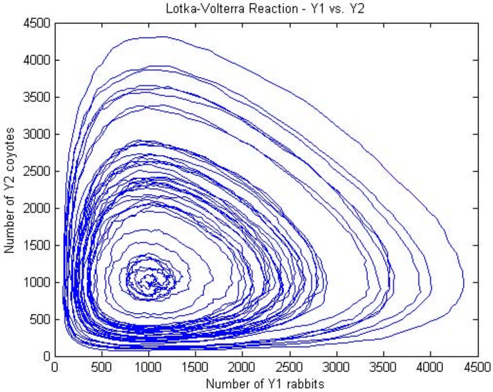
Prey-Predator Systems

- the evolution of a prey-predator systems leads to interesting dynamics
- these dynamics have been encoded in the Lotka-Volterra equation [Solé and Bascompte, 2006]
- depending on the parameters values the system may evolve either to overpopulation, extinction or periodical evolution
- the Lotka-Volterra equation:

$$\frac{dx}{dt} = x(\alpha - \beta y)$$

$$\frac{dy}{dt} = -y(\gamma - \delta x)$$

Lotka-Volterra Equation



A chart depicting the state space defined by the Lotka-Volterra equation.

Synchronised Flashing in Fireflies I

- some species of fireflies have been reported of being able to synchronise their flashing [Camazine et al., 2001]
- synchronous flashing is produced by male during mating
- this synchronisation behaviour is reproducible using simple rules
  - start counting cyclically
  - when perceive a flash, flash and restart counting

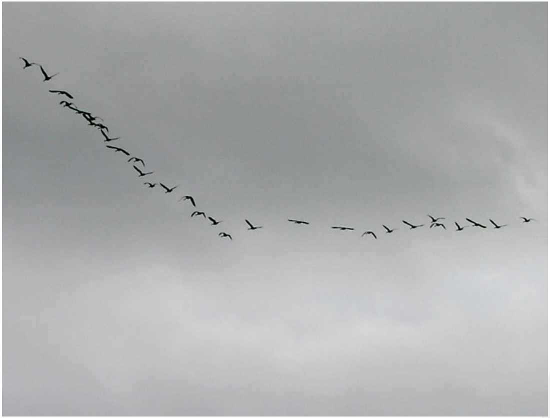
Synchronised Flashing in Fireflies II

A photo of fireflies flashing synchronously.

Schools of Fishes

School of fishes exhibit coordinated swimming: this behaviour can be simulated based on speed, orientation, and distance perception [Camazine et al., 2001].

Flocks of Birds



The picture displays a flock of geese: this behaviour can be simulated based on speed, orientation, and distance perception [Camazine et al., 2001].

Insects Colonies

- behaviours displayed by social insects have always puzzled entomologist
- behaviours such as nest building, sorting, routing were considered requiring elaborated skills
- for instance, termites and ants build very complex nests, whose building criteria are far than trivial, such as inner temperature, humidity, and oxygen concentration

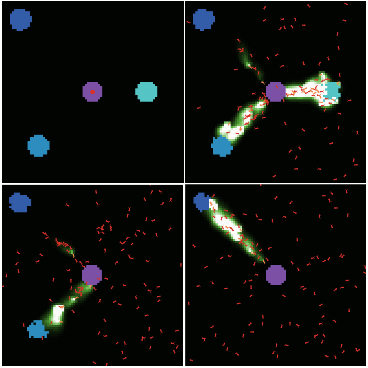
Termites Nest in South Africa

The picture displays the Macrotermes michealseni termite mound of southern Africa.

Trail Formation in Ant Colonies

The picture food foraging ants. When carrying food, ants lay pheromone, adaptively establishing a path between food source and the nest. When sensing pheromone, ants follow the trail to reach the food source.

Simulating Food Foraging



The snapshots display a simulation of food foraging ants featuring a nest and three food sources. Ants find the shortest path to each sources ad consume first the closer sources. When no longer reinforced, the pheromone eventually evaporates.

Focus on. . .

4. Autonomy in Complex Artificial Systems
   - Self-Organisation
   - Self-Organisation and Emergence in Natural Systems
   - Self-Organisation and Emergence in Artificial Systems
   - Self-Organising Systems & Autonomy
   - Multi-Agent Systems vs. Self-Organising Systems
5. Coordination for Self-Organisation & System Autonomy
   - Coordination Models
   - LINDA & Tuple-based Coordination
   - Nature-inspired Coordination
   - Tuple-based models for Nature-inspired Coordination
   - Coordination Technologies

Swarm Intelligence

- *swarm intelligence* is a problem solving approach inspired by collective behaviours displayed by social insects

  [Bonabeau et al., 1999, Bonabeau and Théraulaz, 2000]
- it is not a uniform theory, rather a collection of mechanisms found in natural systems having applications to artificial systems
- applications of swarm intelligence include a variety of problems such as task allocation, routing, synchronisation, sorting
- in swarm intelligence, the most successful initiative is Ant Colony Optimisation

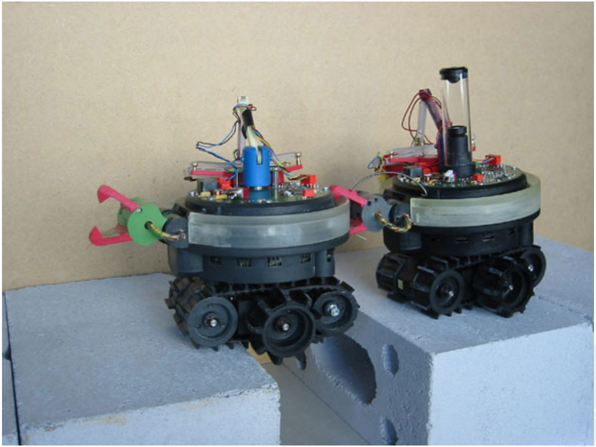
ACO: Ant Colony Optimisation

- ACO [Dorigo and Stützle, 2004] is a population-based metaheuristic that can be used to find approximate solutions to difficult optimisation problems
- a set of software agents called artificial ants search for good solutions to a given optimisation problem
- to apply ACO, the optimisation problem is transformed into the problem of finding the best path on a weighted graph
- ACO provided solutions to problems such as VRP-Vehicle Routing Problem, TSP-Travelling Salesman Problem and packet routing in telecommunication networks

Autonomic Computing

- an industry driven research field initiated by IBM [Kephart and Chess, 2003], mostly motivated by increasing costs in systems maintenance
- basic idea: applying self-organising mechanisms found in human nervous system to develop more robust and adaptive systems
- applications range from a variety of problems such as power saving, security, load balancing

SWARM-BOTS



SWARM-BOTS [Dorigo et al., 2005] was a project funded by European Community tailored to the study of self-organisation and self-assembly of modular robots

AGV – Automated Guided Vehicles

- stigmergy has been successfully applied to several deployments of Automated Guided Vehicles [Weyns et al., 2005, Sauter et al., 2005]
- basically, the AGVs are driven by digital pheromones fields in the same way ants perform food-foraging

Pictures of AGVs

Focus on. . .

Are SOS Autonomous?

- they are *adaptive*, in that they properly respond to external stimuli
- so their autonomy from the environment is *partial*
- at the same time, they are *self-governed*, in that their evolution is self-driven, in some essential sense—it is at least teleonomic
- so, their autonomy is evident, as well

Systems as Agents?

- in the following, we take as understood the fact that the notion of *autonomy* applies to *systems*
- our implicit assumption is that *users* (generally) and *designers* (at some point) consider a *system as a whole*, and conceive it as such
- that is, as a computational system with its own computational autonomy—which for us means an agent, at a certain level of abstraction

→ this basically means that we can evaluate other notions of autonomy for a system as a whole

How Much Autonomy?

- good design of a SOS provides the goals to be achieved, and the means to self-organise the system structure accordingly
- how much autonomy in that?
- how much autonomy from the designer, from the user, from the environment, overall?

Which Autonomy for SOS?

- self-organising systems (SOS) exhibit some autonomy by definition
  - their evolution over time is not pre-defined by the designer
  - → in this sense, SOS are autonomous with respect to the designer
  - however, any evolution of a well-engineered SOS tends towards the tasks / goals assigned by the designer
  - → in this sense, SOS are not autonomous with respect to the designer
  - their evolution over time is not is influenced by the environment, but is not directly driven by it
  - → in this sense, SOS are autonomous with respect to the environment
- most of the SOS we know are natural systems, where it is not clear whether one can say that the goals are somehow self-generated
- however, for sure, computational SOS built from those examples are likely to show executive autonomy, without motivational autonomy

Autonomy of SOS Depends on...

- the models, mechanisms, and technologies adopted for implementing computational SOS

! the level of autonomy of a SOS do not depend straightforwardly on the level of autonomy of the agent components

Component vs. System Autonomy

- SOS are systems with some autonomy made of autonomous components
- however, no clear relationship between the sort of autonomy of components and the sort of autonomy of the system can be stated a priori

→ which basically means that autonomy of a SOS does not necessarily rely upon its components only

→ and also means that issues like responsibly and liability require a non-trivial, non-obvious treatment

Focus on. . .

4. Autonomy in Complex Artificial Systems
   - Self-Organisation
   - Self-Organisation and Emergence in Natural Systems
   - Self-Organisation and Emergence in Artificial Systems
   - Self-Organising Systems & Autonomy
   - Multi-Agent Systems vs. Self-Organising Systems
5. Coordination for Self-Organisation & System Autonomy
   - Coordination Models
   - Linda & Tuple-based Coordination
   - Nature-inspired Coordination
   - Tuple-based models for Nature-inspired Coordination
   - Coordination Technologies

MAS 4 SOS

- is the agent paradigm the right choice for modelling and developing SOS?
- are agents the right abstractions for SOS components?
- are MAS the right way to put together components of a SOS?
- in order to answer this question we have to compare requirements for SOS with features of MAS

SOS Requirements

- from our previous discussion on self-organisation and emergence, a possible basic requirements list can be given as follows:
  - autonomy and encapsulation of behaviour
  - local actions and perceptions
  - distributed environment supporting interactions
  - support for organisation and cooperation concepts

MAS Checklist

It is easy to recognise that the agent paradigm provides suitable abstractions for each aspect

- agents for autonomy and encapsulation of behaviour
- situated agents for local actions and perceptions
- MAS distribution of components, and MAS environment supporting interactions through *coordination*
- MAS support for organisation and cooperation concepts

Next in Line. . .

4. Autonomy in Complex Artificial Systems

5. Coordination for Self-Organisation & System Autonomy

Focus on. . .

4. Autonomy in Complex Artificial Systems
   - Self-Organisation
   - Self-Organisation and Emergence in Natural Systems
   - Self-Organisation and Emergence in Artificial Systems
   - Self-Organising Systems & Autonomy
   - Multi-Agent Systems vs. Self-Organising Systems
5. Coordination for Self-Organisation & System Autonomy
   - **Coordination Models**
   - LINDA & Tuple-based Coordination
   - Nature-inspired Coordination
   - Tuple-based models for Nature-inspired Coordination
   - Coordination Technologies

Interaction & Coordination

Interaction

- most of the complexity of complex computational systems – MAS included – comes from interaction [Omicini et al., 2006]
- along with an essential part of their expressive power [Wegner, 1997]

Coordination

- since coordination is essentially the science of managing the space of interaction [Wegner, 1997]
- coordination models and languages [Ciancarini, 1996] provide abstractions and technologies for the engineering of complex computational systems [Ciancarini et al., 2000]

Coordination Models for Complex Computational Systems

Coordination model as a glue

A coordination model is the glue that binds separate activities into an ensemble [Gelernter and Carriero, 1992]

Coordination model as an agent interaction framework

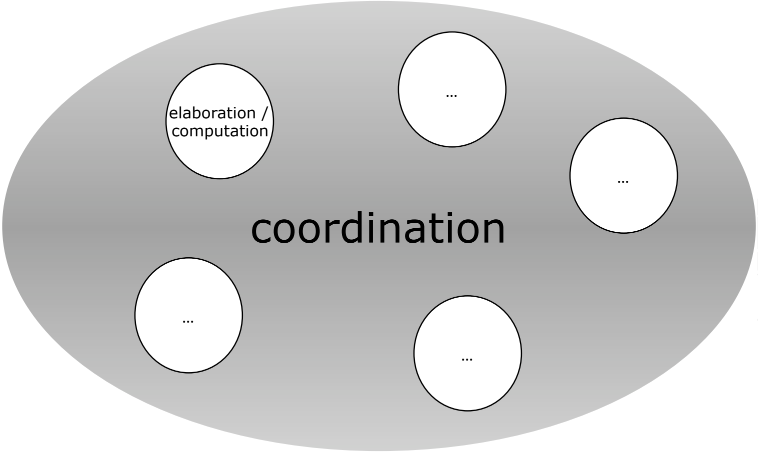
A coordination model provides a framework in which the interaction of active and independent entities called agents can be expressed [Ciancarini, 1996]

Issues for a coordination model

A coordination model should cover the issues of creation and destruction of agents, communication among agents, and spatial distribution of agents, as well as synchronization and distribution of their actions over time [Ciancarini, 1996]

What is Coordination?

Ruling the space of interaction



A New Perspective over Computational Systems

Programming languages

- interaction as an *orthogonal* dimension
- *languages* for interaction / *coordination*

Software engineering

- interaction as an *independent design* dimension
- coordination *patterns*

Artificial intelligence

- interaction as a new *source for intelligence*
- *social intelligence*

A Meta-model for Coordinated Systems I

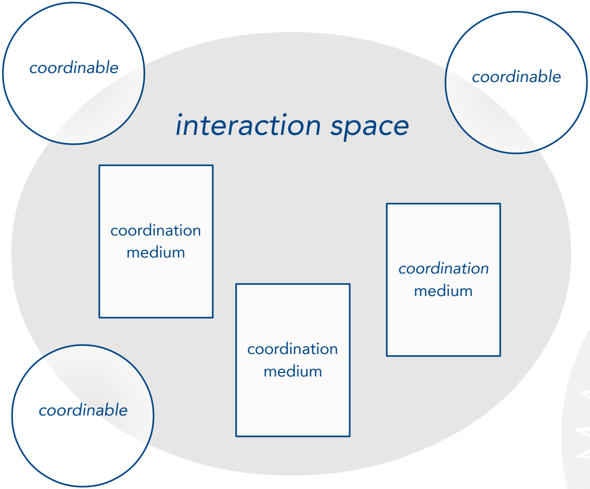
The coordination meta-model [Ciancarini, 1996]

coordination entities — the entities whose mutual interaction is ruled by the model, also called the *coordinables* (or, the *agents*)

coordination media — the abstractions enabling and ruling interaction among coordinables

coordination laws — the rules governing the observable behaviour of coordination media and coordinables, and their interaction as well

A Meta-model for Coordinated Systems II



A Meta-model for Coordinated Systems III

The coordination media. . .

- “fill” the interaction space
- enable / promote / govern the admissible / desirable / required interactions among the interacting entities
- according to some *coordination laws*
  - enacted by the behaviour of the media
  - defining the semantics of coordination

Focus on. . .

The Ancestor

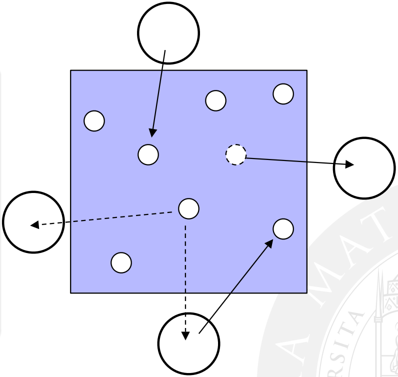
Linda [Gelernter, 1985]

- Linda is the ancestor of all tuple-based coordination models [Rossi et al., 2001]
- in Linda, agents synchronise, cooperate, compete
  - based on tuples
  - available in the tuple spaces, working as the coordination media
  - by *associatively* accessing, consuming and producing tuples
- the same holds for any tuple-based coordination model

The Tuple-space Meta-model

The basics [Gelernter, 1985]

- *coordinables* synchronise, cooperate, compete
  - based on *tuples*
  - available in the *tuple space*
  - by *associatively* accessing, consuming and producing tuples



Tuple-based / Space-based Coordination Systems

LINDA meta-model [Ciancarini, 1996]

coordination media tuple spaces

- as multiset / bag of data objects / structures called *tuples*

communication language tuples / tuple templates

tuples as ordered collections of (possibly heterogeneous) information items

templates as specifications of tuple sets

coordination language tuple space primitives

- as a set of operations to put, browse and retrieve tuples to/from the space

Multiple Tuple Spaces

`ts ? out(T)`

- LINDA tuple space might be a bottleneck for coordination
- many extensions have focussed on making a multiplicity of tuple spaces available to processes
  - each of them encapsulating a portion of the coordination load
  - either hosted by a single machine, or distributed across the network
- syntax required, and dependent on particular models and implementations
  - a space for tuple space names, possibly including network location
  - operators to associate LINDA operators to tuple spaces
- for instance, `ts @ node ? out(p)` may denote the invocation of operation `out(p)` over tuple space `ts` on node `node`

Main Features of Tuple-based Coordination

Main features of the LINDA model

tuples a tuple is an ordered collection of knowledge chunks, possibly heterogeneous in sort

generative communication until explicitly withdrawn, the tuples generated by coordinables have an independent existence in the tuple space; a tuple is equally accessible to all the coordinables, but is bound to none

associative access templates allow accessing tuples through their content & structure, rather than by name, address, or location

suspensive semantics operations may be suspended based on unavailability of matching tuples, and be woken up when such tuples become available

Focus on. . .

4. Autonomy in Complex Artificial Systems
   - Self-Organisation
   - Self-Organisation and Emergence in Natural Systems
   - Self-Organisation and Emergence in Artificial Systems
   - Self-Organising Systems & Autonomy
   - Multi-Agent Systems vs. Self-Organising Systems
5. Coordination for Self-Organisation & System Autonomy
   - Coordination Models
   - LINDA & Tuple-based Coordination
   - **Nature-inspired Coordination**
   - Tuple-based models for Nature-inspired Coordination
   - Coordination Technologies

Nature-inspired Coordination for MAS

Coordination issues in natural systems

- coordination issues did not first emerge in computational systems
- [Grassé, 1959] noted that in termite societies *"The coordination of tasks and the regulation of constructions are not directly dependent from the workers, but from constructions themselves."*

Coordination as the key issue

- many well-known examples of natural systems – and, more generally, of complex systems – seemingly rely on simple yet powerful coordination mechanisms for their key features—such as self-organisation
- it makes sense to focus on nature-inspired coordination models as the core of complex nature-inspired MAS

Stigmergy I

Stigmergy in insect societies

- nature-inspired models of coordination are grounded in studies on the behaviour of social insects, like ants or termites
- [Grassé, 1959] introduced the notion of stigmergy as the fundamental coordination mechanism in termite societies
- in ant colonies, pheromones act as environment markers for specific social activities, and drive both the *individual* and the *social* behaviour of ants

Stigmergy II

Stigmergy in computational systems

- nowadays, stigmergy generally refers to a set of nature-inspired coordination mechanisms mediated by the *environment*
- *digital pheromones* [Parunak et al., 2002] and other *signs* made and sensed in a shared environment [Parunak, 2006] can be exploited for the engineering of adaptive and self-organising MAS

Chemical Coordination

Chemical reactions as (natural) coordination laws

- inspiration comes from the idea that complex physical phenomena are driven by the (relatively) simple chemical reactions
- coordinating the behaviours of a huge amount of agents, as well as the global system evolution

Chemical reactions as (computational) coordination laws

- Gamma [Banâtre and Le Métayer, 1990] is a *chemistry-inspired coordination* model—as for the CHAM (chemical abstract machine) model [Berry, 1992]
- coordination in Gamma is conceived as the evolution of a space governed by chemical-like rules, globally working as a rewriting system [Banătre et al., 2001]

Field-based Coordination

Computational fields as coordination laws

- field-based coordination models like Co-Fields [Mamei and Zambonelli, 2006] are inspired by the way masses and particles move and self-organise according to gravitational/electromagnetic fields
- there, computational force fields – generated either by the mobile agents or by the pervasive coordination infrastructure – propagate across the environment, and drive the actions and motion of the agent themselves

(Bio)chemical Coordination

Chemical reactions as coordination laws

- chemical tuple spaces [Viroli et al., 2010] exploit the chemical metaphor at its full extent—beyond Gamma
- data, devices, and software agents are represented in terms of chemical reactants, and system behaviour is expressed by means of chemical-like laws
- which are actually time-dependent and stochastic
- embedded within the coordination medium
- biochemical tuple spaces [Viroli and Casadei, 2009] add *compartments*, *diffusion*, and *stochastic behaviour* of coordination primitives

Basic Issues of Nature-inspired Coordination I

Environment

Environment is essential in nature-inspired coordination

- it works as a mediator for agent interaction — through which agents can communicate and coordinate indirectly
- it is active — featuring autonomous dynamics, and affecting agent coordination
- it has a structure — requiring a notion of *locality*, and allowing agents of any sort to *move* through a topology

Basic Issues of Nature-inspired Coordination II

Stochastic behaviour

Complex systems typically require probabilistic models

- *don't know* / *don't care* non-deterministic mechanisms are not expressive enough to capture all the properties of complex systems such as biochemical and social systems
- probabilistic mechanisms are required to fully capture the dynamics of coordination in nature-inspired systems
- coordination models should feature (possibly simple yet) expressive mechanisms to provide coordinated systems with stochastic behaviours

Focus on. . .

4. Autonomy in Complex Artificial Systems
   - Self-Organisation
   - Self-Organisation and Emergence in Natural Systems
   - Self-Organisation and Emergence in Artificial Systems
   - Self-Organising Systems & Autonomy
   - Multi-Agent Systems vs. Self-Organising Systems
5. Coordination for Self-Organisation & System Autonomy
   - Coordination Models
   - LINDA & Tuple-based Coordination
   - Nature-inspired Coordination
   - **Tuple-based models for Nature-inspired Coordination**
   - Coordination Technologies

Linda is *not* a Nature-inspired Model

Warning

Linda is *not* a Nature-inspired Model

So, *why* Linda?

Why tuple-based models?

Why Tuple-based Models? I

Expressiveness

- LINDA is a sort of *core* coordination model
- making it easy to face and solve many typical problems of complex distributed systems
- *complex* coordination problems are solved with *few*, *simple* primitives
- whatever the model used to measure expressiveness of coordination, tuple-based languages are highly-expressive [Busi et al., 1998]

Why Tuple-based Models? II

Environment-based coordination

- generative communication [Gelernter, 1985] requires *permanent* coordination abstractions
- so, the *coordination infrastructure* provides agents with tuple spaces as coordination services
    - *coordination as a service* (CaaS) [Viroli and Omicini, 2006]
- they can be interpreted as coordination artefacts shaping computational *environment* [Omicini et al., 2004]
    - and used with different levels of awareness by both intelligent and "stupid" agents [Omicini, 2013]
- as such, they can be exploited to support environment-based coordination [Ricci et al., 2005]

Why Tuple-based Models? III

Extensibility

- whatever its expressiveness, LINDA was conceived as a coordination model for closed, parallel systems
- so, in fact, some relevant problems of today open, concurrent systems cannot be easily solved with LINDA either in practice or in theory
- as a result, tuple-based models have been extended with new simple yet powerful mechanisms
- generating a plethora of tuple-based coordination models [Rossi et al., 2001]

Why Tuple-based Models? IV

Nature-inspired extensions

- LINDA may *not* be nature-inspired, but many of its extensions *are*
- many of the coordination models depicted before
  - stigmergy [Parunak, 2006]
  - field-based [Mamei and Zambonelli, 2004]
  - chemical [Viroli et al., 2010] and biochemical [Viroli and Casadei, 2009]
- along with many others, such as
  - cognitive stigmergy [Ricci et al., 2007]
  - pervasive ecosystems [Viroli et al., 2012]
- are actually nature-inspired tuple-based coordination models

Examples I

StoKlaim

- StoKlaim [De Nicola et al., 2006] – a *stochastic* extension of the Linda-derived Klaim model for mobile coordination [De Nicola et al., 1998] – adds distribution rates to coordination primitives—thus making it possible the modelling of non-deterministic real-life phenomena such as failure rates and inter-arrival times

SwarmLinda

- SwarmLinda [Tolksdorf and Menezes, 2004] enhances Linda implementation with swarm intelligence to achieve features such as scalability, adaptiveness, and fault-tolerance—by modelling tuple templates as ants, featuring probabilistic behaviour when looking for matching tuples in a distributed setting

Examples II

Situated ReSpecT [Mariani and Omicini, 2013]

- ReSpecT [Omicini and Denti, 2001] generally addresses situated dependency by capturing *time*, *space*, and *environment events*, and supporting the definition and enforcement of *situated coordination policies*
- so, ReSpecT-programmed tuple centres can work as situated abstractions for MAS-environment coordination

Blending Metaphors

Mixing abstractions & mechanisms from different conceptual sources

- most natural systems, when observed in their whole complexity, exhibit *layers* each one featuring its own metaphors and mechanisms
- correspondingly, many novel approaches to complex MAS coordination integrate diverse sources of inspiration, e.g.:
  - TOTA [Mamei and Zambonelli, 2004] exploits mechanisms from both stigmergic and field-based coordination
  - the SAPERE coordination model for pervasive service ecosystems [Zambonelli et al., 2011, Viroli et al., 2012] integrates
    - the *chemical* metaphor for driving the evolution of coordination abstractions
    - *biochemical* abstractions for topology and diffusion
    - the notion of *ecosystem* in order to model the overall system structure and dynamics

Expressing Full Dynamics

Expressing the *full dynamics* of complex natural systems

- mostly, coordination models just capture *some* of the overall system dynamics
- which makes them basically *fail*
  - for instance, Gamma mimics chemical reactions, but does not capture essential issues in chemical processes such as reaction rates and concentration [Banâtre and Le Métayer, 1990, Banǎtre et al., 2001]
  - instead, *(bio)chemical tuple spaces* fully exploit the chemical metaphor by providing time-dependent and stochastic chemical laws [Viroli et al., 2010, Viroli and Casadei, 2009]
- more generally, the goal is to allow coordinated MAS to capture and express the full dynamics of complex natural systems

Core Mechanisms I

Understanding the *basic elements* of *expressiveness*

- LINDA is a glaring example of a minimal set of coordination mechanisms providing a wide range of coordination behaviours
- the goal is understanding the minimal set of coordination primitives required to design complex stochastic behaviours
- for instance, *uniform coordination primitives* – that is, LINDA-like coordination primitives returning tuples matching a template with a uniform distribution [Gardelli et al., 2007] – seemingly capture the full-fledged dynamics of real chemical systems within the coordination abstractions

Core Mechanisms II

Issues

- autonomy and the limits of computational systems
  - *expressiveness* of languages and technologies
  - (*technically*, *ethically*, *legally*) *admissible behaviours* of computational systems

Predicting Complex Behaviours I

Engineering unpredictable systems around predictable abstractions

- coordination models are meant to harness the complexity of complex MAS [Ciancarini et al., 2000]
- coordination abstractions are often *at the core* of complex MAS
- while this does not make complex MAS generally predictable, it makes it possible in principle to make them *partially predictable*, based on the predictably of the core coordinative behaviour
- suitably-formalised coordination abstractions, along with a suitably-defined engineering methodology, could in principle ensure the predictability of given MAS properties within generally-unpredictable MAS

Predicting Complex Behaviours II

Issues

- autonomy and predictability
  - is unpredictability a pre-condition to choice, freedom, and so autonomy?
  - how could unpredictability coexist with well-founded notions of responsibility and liability?
- partial predictability?

Coordination for Simulation I

Simulation of complex systems is a multidisciplinary issue

- ... ranging from physics to biology, from economics to social sciences
- no complex system of any sort can be studied nowadays without the support of suitable simulation tools
- nowadays, experiments done *in silico* are at least as relevant as those *in vitro* and *in vivo*

Coordination for Simulation II

Interaction issues are prominent in complex systems

- coordination technologies potential core of agent-based simulation frameworks
- in particular, self-organising nature-inspired coordination models are well suited for the simulation of complex systems
- so, coordination middleware could play a central role in the development of rich agent-based simulation frameworks for complex systems
- e.g., [González Pérez et al., 2013]

Issues

- autonomy and simulation
  - what autonomy is required to simulate autonomous systems?

Knowledge-oriented Coordination I

Integrating nature-inspired with knowledge-oriented coordination

- intelligent MAS in knowledge intensive environments – as well as complex socio-technical systems, in general – require automatic understanding of data and information
- knowledge-oriented coordination exploits coordination abstractions enriched so as to allow for semantic interpretation by intelligent agents [Fensel, 2004, Nardini et al., 2013]
- for instance
  - chemical tuple spaces
  - SAPERE coordination abstractions and mechanisms
  - *semantic tuple centres* [Nardini et al., 2011]

all relay on the semantic interpretation of coordination items

Knowledge-oriented Coordination II

Self-organisation of knowledge

- explicit search of information is going to become ineffective while the amount of available knowledge grows at incredible rates
- knowledge should autonomously organise and flow from producers to consumers
- knowledge self-organisation for knowledge-intensive MAS

Knowledge-oriented Coordination III

MoK (Molecules of Knowledge) [Mariani and Omicini, 2012a]

- Molecules of Knowledge is a a nature-inspired coordination model promoting knowledge self-organisation, where
  - sources of knowledge continuously produce and inject *atoms of knowledge* in biochemical compartments
  - knowledge atoms may then aggregate in *molecules* and diffuse
  - knowledge producers, managers and consumers are modelled as *catalysts*, whose workspaces are biochemical compartments, and their knowledge-oriented actions become enzymes influencing atoms aggregation and molecules diffusion
  - so as to make relevant knowledge spontaneously aggregate and autonomously move towards potentially interested knowledge workers
- the first application scenario for experimenting with MoK is *news management* [Mariani and Omicini, 2012b]

Knowledge-oriented Coordination IV

Issues

- autonomy and knowledge
  - if (informed) choice is essential to freedom, and freedom is essential to autonomous choice, then *knowledge* is *essential to autonomy*
- autonomy of knowledge?
  - what is autonomy, when *knowledge* chunks *autonomously move* towards knowledge consumers?

Focus on...

4. Autonomy in Complex Artificial Systems
   - Self-Organisation
   - Self-Organisation and Emergence in Natural Systems
   - Self-Organisation and Emergence in Artificial Systems
   - Self-Organising Systems & Autonomy
   - Multi-Agent Systems vs. Self-Organising Systems
5. Coordination for Self-Organisation & System Autonomy
   - Coordination Models
   - LINDA & Tuple-based Coordination
   - Nature-inspired Coordination
   - Tuple-based models for Nature-inspired Coordination
   - **Coordination Technologies**

Coordination Middleware

JavaSpaces `http://www.oracle.com/technetwork/articles/javase/javaspaces-140665.html`
: A Java-based high-level tool for building distributed and collaborative applications [Freeman et al., 1999]

Law-Governed Interaction (LGI) `http://www.moses.rutgers.edu`
: A decentralised coordination and control mechanism for distributed systems [Minsky and Ungureanu, 2000]

TuCSoN `http://tucson.unibo.it`
: A model and technology for tuple-based coordination of complex distributed systems [Omicini and Zambonelli, 1999]

Part III

Bibliography

References I

Ashby, W. R. (1947).
Principles of self-organizing dynamic systems.
Journal of General Psychology, 37:125–128.

Banătre, J.-P., Fradet, P., and Le Métayer, D. (2001).
Gamma and the chemical reaction model: Fifteen years after.
In Calude, C. S., Păun, G., Rozenberg, G., and Salomaa, A., editors, *Multiset Processing. Mathematical, Computer Science, and Molecular Computing Points of View*, volume 2235 of *LNCS*, pages 17–44. Springer.

Banâtre, J.-P. and Le Métayer, D. (1990).
The GAMMA model and its discipline of programming.
Science of Computer Programming, 15(1):55–77.

Berry, G. (1992).
The chemical abstract machine.
Theoretical Computer Science, 96(1):217–248.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999).
Swarm Intelligence: From Natural to Artificial Systems.
Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, 198 Madison Avenue, New York, New York 10016, United States of America.

References II

Bonabeau, E. and Théraulaz, G. (2000).
Swarm smarts (behavior of social insects as model for complex systems).
Scientific American, 282(3):72–79.

Bratman, M. E. (1987).
Intention, Plans, and Practical Reason.
Harvard University Press.

Bratman, M. E. (1990).
What is intention?
In Cohen, P. R., Morgan, J. L., and Pollack, M. E., editors, *Intentions in Communication*, pages 15–32. The MIT Press, Cambridge, MA.

Busi, N., Gorrieri, R., and Zavattaro, G. (1998).
A process algebraic view of Linda coordination primitives.
Theoretical Computer Science, 192(2):167–199.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001).
Self-Organization in Biological Systems.
Princeton Studies in Complexity. Princeton University Press, 41 William Street, Princeton, New Jersey 08540, United States of America.

References III

Castelfranchi, C. (1995).
Guarantees for autonomy in cognitive agent architecture.
In Wooldridge, M. J. and Jennings, N. R., editors, *Intelligent Agents*, volume 890 of *Lecture Notes in Computer Science*, pages 56–70. Springer Berlin Heidelberg.
ECAI-94 Workshop on Agent Theories, Architectures, and Languages (ATAL) Amsterdam, The Netherlands 8–9 August 1994. Proceedings.

Ciancarini, P. (1996).
Coordination models and languages as software integrators.
ACM Computing Surveys, 28(2):300–302.

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.
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.

References IV

Dasgupta, A. and Ghose, A. K. (2011).
BDI agents with objectives and preferences.
In Omicini, A., Sardina, S., and Vasconcelos, W., editors, *Declarative Agent Languages and Technologies VIII*, volume 6619 of *Lecture Notes in Computer Science*, pages 22–39. Springer Berlin Heidelberg.

De Nicola, R., Ferrari, G., and Pugliese, R. (1998).
KLAIM: A kernel language for agent interaction and mobility.
IEEE Transaction on Software Engineering, 24(5):315–330.

De Nicola, R., Latella, D., Katoen, J.-P., and Massink, M. (2006).
StoKlaim: A stochastic extension of Klaim.
Technical Report 2006-TR-01, Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo" (ISTI).

De Wolf, T. and Holvoet, T. (2005).
Emergence versus self-organisation: Different concepts but promising when combined.
In Brueckner, S., Di Marzo Serugendo, G., Karageorgos, A., and Nagpal, R., editors, *Engineering Self Organising Systems: Methodologies and Applications*, volume 3464 of *LNCS (LNAI)*, pages 1–15. Springer.

References V

Dennett, D. (1971).
Intentional systems.
Journal of Philosophy, 68:87–106.

Dennett, D. (2007).
Intentional systems theory.
In Beckermann, A. and Walter, S., editors, *Oxford Handbook of the Philosophy of Mind*, chapter 19. Oxford University Press.

Di Marzo Serugendo, G., Gleizes, M.-P., and Karageorgos, A. (2011).
Self-organising software.
In Di Marzo Serugendo, G., Gleizes, M.-P., and Karageorgos, A., editors, *Self-organising Software. From Natural to Artificial Adaptation*, Natural Computing Series, chapter 2, pages 7–32. Springer Berlin Heidelberg, Berlin, Heidelberg.

Dorigo, M. and Stützle, T. (2004).
Ant Colony Optimization.
MIT Press, Cambridge, MA.

References VI

Dorigo, M., Tuci, E., Mondada, F., Nolfi, S., Deneubourg, J.-L., Floreano, D., and Gambardella, L. M. (2005).
The SWARM-BOTS project.
Künstliche Intelligenz, 4/05:32–35.
Also available at http://www.swarm-bots.org as IRIDIA Technical Report No. TR/IRIDIA/2005-018.

Edwards, G. (2013).
Military autonomous & robotic systems. considerations for the way forward from a UK military perspective.
Air Power Review, 16(3):50–71.

Fensel, D. (2004).
Triple-space computing: Semantic web services based on persistent publication of information.
In Aagesen, F. A., Anutariya, C., and Wuwongse, V., editors, *Intelligence in Communication Systems*, volume 3283 of *LNCS*, pages 43–53.
IFIP International Conference (INTELLCOMM 2004), Bangkok, Thailand, 23–26 November 2004. Proceedings.

References VII

Freeman, E., Hupfer, S., and Arnold, K. (1999).
JavaSpaces Principles, Patterns, and Practice: Principles, Patterns and Practices.
The Jini Technology Series. Addison-Wesley Longman.

Gardelli, L., Viroli, M., Casadei, M., and Omicini, A. (2007).
Designing self-organising MAS environments: The collective sort case.
In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNAI*, pages 254–271. Springer.
3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.

Gelernter, D. (1985).
Generative communication in Linda.
ACM Transactions on Programming Languages and Systems, 7(1):80–112.

Gelernter, D. and Carriero, N. (1992).
Coordination languages and their significance.
Communications of the ACM, 35(2):97–107.

Goldstein, J. (1999).
Emergence as a construct: History and issues.
Emergence, 1(1):49–72.

References VIII

González Pérez, P. P., Omicini, A., and Sbaraglia, M. (2013).
A biochemically-inspired coordination-based model for simulating intracellular signalling pathways.
Journal of Simulation.
Special Issue on Agent-based Modeling and Simulation.

Gould, S. J. (2002).
The Structure of Evolutionary Theory.
The Belknap Press of Harvard University Press.

Graesser, S. F. A. (1996).
Is it an agent, or just a program?: A taxonomy for autonomous agents.
In Müller, J. P., Wooldridge, M. J., and Jennings, N. R., editors, *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.

Grassé, P.-P. (1959).
La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs.
Insectes Sociaux, 6(1):41–80.

References IX

Kephart, J. O. and Chess, D. M. (2003).
The vision of autonomic computing.
Computer, 36(1):41–50.

Kirschner, M. W. and Gerhart, J. C. (2006).
The Plausibility of Life: Resolving Darwin's Dilemma.
Yale Universtiy Press.

Kitano, H. (2002).
Foundations of Systems Biology.
MIT Press.

Liu, J. and Tsui, K. C. (2006).
Toward nature-inspired computing.
Communications of the ACM, 49(10):59–64.

Mamei, M. and Zambonelli, F. (2004).
Programming pervasive and mobile computing applications with the TOTA middleware.
In *Pervasive Computing and Communications*, pages 263–273.
2nd IEEE Annual Conference (PerCom 2004), Orlando, FL, USA, 14–17 March 2004.
Proceedings.

References X

Mamei, M. and Zambonelli, F. (2006).
Field-Based Coordination for Pervasive Multiagent Systems. Models, Technologies, and Applications.
Springer Series in Agent Technology. Springer.

Mariani, S. and Omicini, A. (2012a).
Molecules of Knowledge: Self-organisation in knowledge-intensive environments.
In Fortino, G., Bădică, C., Malgeri, M., and Unland, R., editors, *Intelligent Distributed Computing VI*, volume 446 of *Studies in Computational Intelligence*, pages 17–22. Springer.
6th International Symposium on Intelligent Distributed Computing (IDC 2012), Calabria, Italy, 24-26 September 2012. Proceedings.

Mariani, S. and Omicini, A. (2012b).
Self-organising news management: The *Molecules of Knowledge* approach.
In Fernandez-Marquez, J. L., Montagna, S., Omicini, A., and Zambonelli, F., editors, *1st International Workshop on Adaptive Service Ecosystems: Natural and Socially Inspired Solutions (ASENSIS 2012)*, pages 11–16, SASO 2012, Lyon, France.
Pre-proceedings.

References XI

Mariani, S. and Omicini, A. (2013).
Event-driven programming for situated MAS with ReSpecT tuple centres.
In Klusch, M., Thimm, M., and Paprzycki, M., editors, *Multiagent System Technologies*, volume 8076 of *LNAI*, pages 306–319. Springer.
11th German Conference (MATES 2013), Koblenz, Germany, 16-20 September 2013. Proceedings.

Maturana, H. R. and Varela, F. G. (1980).
Autopoiesis and Cognition: The Realization of the Living, volume 42 of *Boston Studies in the Philosophy of Science*.
D. Reidel Publishing Company, Dordrecht, Holland.

Mazal, Z., Kočí, R., Janoušek, V., and Zbořil, F. (2008).
PNagent: A framework for modelling BDI agents using object oriented Petri nets.
In *2008 8th International Conference on Intelligent Systems Design and Applications (ISDA 2008)*, volume 2, pages 420–425. IEEE.

McCarthy, J. (1979).
Ascribing mental qualities to machines.
In Ringle, M., editor, *Philosophical Perspectives in Artificial Intelligence*, Harvester Studies in Cognitive Science, pages 161–195. Harvester Press, Brighton.

References XII

Minsky, N. H. and Ungureanu, V. (2000).
Law-Governed interaction: A coordination and control mechanism for heterogeneous distributed systems.
ACM Transactions on Software Engineering and Methodology (TOSEM), 9(3):273–305.

Nardini, E., Omicini, A., and Viroli, M. (2013).
Semantic tuple centres.
Science of Computer Programming.
Special Issue on Self-Organizing Coordination. To appear.

Nardini, E., Omicini, A., Viroli, M., and Schumacher, M. I. (2011).
Coordinating e-health systems with TuCSoN semantic tuple centres.
Applied Computing Review, 11(2):43–52.

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

References XIII

Omicini, A. (2013).
Agents writing on walls: Cognitive stigmergy and beyond.
In Paglieri, F., Tummolini, L., Falcone, R., and Miceli, M., editors, *The Goals of Cognition. Essays in Honor of Cristiano Castelfranchi*, chapter 29, pages 543–556. College Publications, London.

Omicini, A. and Denti, E. (2001).
From tuple spaces to tuple centres.
Science of Computer Programming, 41(3):277–294.

Omicini, A., Ricci, A., and Viroli, M. (2006).
The multidisciplinary patterns of interaction from sciences to Computer Science.
In Goldin, D. Q., Smolka, S. A., and Wegner, P., editors, *Interactive Computation: The New Paradigm*, pages 395–414. Springer.

Omicini, A., Ricci, A., Viroli, M., Castelfranchi, C., and Tummolini, L. (2004).
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.

References XIV

Omicini, A. and Zambonelli, F. (1999).
Coordination for Internet application development.
Autonomous Agents and Multi-Agent Systems, 2(3):251–269.
Special Issue: Coordination Mechanisms for Web Agents.

Parunak, H. V. D. (2006).
A survey of environments and mechanisms for human-human stigmergy.
In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for Multi-Agent Systems II*, volume 3830 of *LNCS*, pages 163–186. Springer.

Parunak, H. V. D., Brueckner, S., and Sauter, J. (2002).
Digital pheromone mechanisms for coordination of unmanned vehicles.
In Castelfranchi, C. and Johnson, W. L., editors, *1st International Joint Conference on Autonomous Agents and Multiagent systems*, volume 1, pages 449–450, New York, NY, USA. ACM.

Rao, A. S. and Georgeff, M. P. (1992).
An abstract architecture for rational agents.
In Nebel, B., Rich, C., and Swartout, W. R., editors, *3rd International Conference on Principles of Knowledge Representation and Reasoning (KR '92)*, pages 439–449, Cambridge, MA, USA. Morgan Kaufmann.
Proceedings.

References XV

Rao, A. S. and Georgeff, M. P. (1995).
BDI agents: From theory to practice.
In Lesser, V. R. and Gasser, L., editors, *1st International Conference on Multi Agent Systems (ICMAS 1995)*, pages 312–319, San Francisco, CA, USA. The MIT Press.

Rao, A. S. and Georgeff, M. P. (1998).
Decision procedures for BDI logics.
Journal of Logic and Computation, 8(3):293–342.

Ricci, A., Omicini, A., Viroli, M., Gardelli, L., and Oliva, E. (2007).
Cognitive stigmergy: Towards a framework based on agents and artifacts.
In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNCS*, pages 124–140. Springer.
3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.

Ricci, A., Viroli, M., and Omicini, A. (2005).
Environment-based coordination through coordination artifacts.
In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for Multi-Agent Systems*, volume 3374 of *LNAI*, pages 190–214. Springer.
1st International Workshop (E4MAS 2004), New York, NY, USA, 19 July 2004. Revised Selected Papers.

References XVI

Rossi, D., Cabri, G., and Denti, E. (2001).
Tuple-based technologies for coordination.
In Omicini, A., Zambonelli, F., Klusch, M., and Tolksdorf, R., editors, *Coordination of Internet Agents: Models, Technologies, and Applications*, chapter 4, pages 83–109. Springer.

Rosslenbroich, B. (2014).
On the Origin of Autonomy. A New Look at the Major Transitions in Evolution.
History, Philosophy and Theory of the Life Sciences. Springer International Publishing.

Sartor, G. and Omicini, A. (2016).
The autonomy of technological systems and responsibilities for their use.
In Bhuta, N., Beck, S., Geiß, R., Liu, H.-Y., and Kreß, C., editors, *Autonomous Weapon Systems. Law, Ethics, Policy*, chapter 3, pages 39–74. Cambridge University Press, Cambridge, UK.

References XVII

Sauter, J. A., Matthews, R. S., Parunak, H. V. D., and Brueckner, S. (2005).
Proceedings of the 4th international joint conference on autonomous agents and multiagent systems (aamas 2005).
In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M. P., and Wooldridge, M., editors, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 903–910, Utrecht, The Netherlands. ACM Press.

Simon, H. A. (1962).
The architecture of complexity.
Proceedings of the American Philosophical Society, 106(6):467–482.

Solé, R. V. and Bascompte, J. (2006).
Self-Organization in Complex Ecosystems.
Number 42 in Monographs in population Biology. Princeton University Press, 41 William Street, Princeton, New Jersey 08540, United States of America.

Thompson, E. (2010).
Mind in Life: Biology, Phenomenology, and the Sciences of Mind.
Belknap Press.

References XVIII

Tolksdorf, R. and Menezes, R. (2004).
Using Swarm Intelligence in Linda Systems.
In Omicini, A., Petta, P., and Pitt, J., editors, *Engineering Societies in the Agents World IV*, volume 3071 of *LNCS*, pages 49–65. Springer.
4th International Workshops (ESAW 2003), London, UK, 29-31 October 2003. Revised Selected and Invited Papers.

Varela, F. G., Maturana, H. R., and Uribe, R. (1974).
Autopoiesis: The organization of living systems, its characterization and a model.
Biosystems, 5(4):187–196.

Viroli, M. and Casadei, M. (2009).
Biochemical tuple spaces for self-organising coordination.
In Field, J. and Vasconcelos, V. T., editors, *Coordination Languages and Models*, volume 5521 of *LNCS*, pages 143–162. Springer, Lisbon, Portugal.
11th International Conference (COORDINATION 2009), Lisbon, Portugal, June 2009. Proceedings.

References XIX

Viroli, M., Casadei, M., Nardini, E., and Omicini, A. (2010).
Towards a chemical-inspired infrastructure for self-* pervasive applications.
In Weyns, D., Malek, S., de Lemos, R., and Andersson, J., editors, *Self-Organizing Architectures*, volume 6090 of *LNCS*, chapter 8, pages 152–176. Springer.
1st International Workshop on Self-Organizing Architectures (SOAR 2009), Cambridge, UK, 14-17 September 2009, Revised Selected and Invited Papers.

Viroli, M. and Omicini, A. (2006).
Coordination as a service.
Fundamenta Informaticae, 73(4):507–534.
Special Issue: Best papers of FOCLASA 2002.

Viroli, M., Pianini, D., Montagna, S., and Stevenson, G. (2012).
Pervasive ecosystems: a coordination model based on semantic chemistry.
In Ossowski, S., Lecca, P., Hung, C.-C., and Hong, J., editors, *27th Annual ACM Symposium on Applied Computing (SAC 2012)*, Riva del Garda, TN, Italy. ACM.

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

References XX

Weyns, D., Schelfthout, K., Holvoet, T., and Lefever, T. (2005).
Proceedings of the 4th international joint conference on autonomous agents and multiagent systems (aamas 2005).
In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M. P., and Wooldridge, M., editors, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 67–74, Utrecht, The Netherlands. ACM Press.

Whitworth, B. (2006).
Socio-technical systems.
In Ghaou, C., editor, *Encyclopedia of Human Computer Interaction*, pages 533–541. IGI Global.

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

References XXI

Zambonelli, F., Castelli, G., Ferrari, L., Mamei, M., Rosi, A., Di Marzo, G., Risoldi, M., Tchao, A.-E., Dobson, S., Stevenson, G., Ye, Y., Nardini, E., Omicini, A., Montagna, S., Viroli, M., Ferscha, A., Maschek, S., and Wally, B. (2011).
Self-aware pervasive service ecosystems.
Procedia Computer Science, 7:197–199.
Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11).

Frontiers of Autonomous Systems

Andrea Omicini
andreaomicini.apice.unibo.it
andrea.omicini@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum—Università di Bologna, Italy

Collegio Superiore
Bologna, Italy
March 2018