Nature-Inspired Coordination & Self-Organisation

Autonomous Systems

Sistemi Autonemi

Andrea Omicini

andrea.omicini@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)

Alma Mater Studiorum – Università di Bologna

Academic Year 2014/2015
1. Nature-inspired Coordination

2. Examples

3. Tuples

4. Trends
Outline

1. Nature-inspired Coordination
2. Examples
3. Tuples
4. Trends
Nature-inspired Coordination

Nature-inspired Models for SOS

Complex natural systems

- such as physical, chemical, biochemical, biological, social systems
- natural system exhibit features
  - such as distribution, openness, 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
Multi-Agent Systems for SOS

**MAS as complex systems [Omicini and Zambonelli, 2004]**

- Agents as sources of complexity
  - **Autonomy**: Unpredictable behaviour
  - **Sociality**: Non-compositional behaviours
  - **Situatedness**: Unpredictable interaction with the environment
- Multi-Agent Systems (MAS) as sources of complexity
  - Multiplicity of interacting components
  - Global vs. local structure and behaviour—macro vs. micro level

**MAS for complex systems [Zambonelli and Omicini, 2004]**

MAS as tools for

- Modelling complex systems
- Engineering complex system
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]
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.
Outline

1. Nature-inspired Coordination
2. Examples
3. Tuples
4. Trends
Outline

1. Nature-inspired Coordination
2. Examples
   - Early
   - Modern
   - Issues
3. Tuples
4. Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - Coordination & Stochastic Systems
   - Full Dynamics
   - Core Mechanisms
   - Blending Metaphors
   - Predicting Complex Behaviours
   - Knowledge-oriented Coordination
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 [Banâtre et al., 2001]
1. Nature-inspired Coordination
2. Examples
   - Early
   - Modern
   - Issues
3. Tuples
4. Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - Coordination & Stochastic Systems
   - Full Dynamics
   - Core Mechanisms
   - Blending Metaphors
   - Predicting Complex Behaviours
   - Knowledge-oriented Coordination
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
Outline

1 Nature-inspired Coordination
2 Examples
   • Early
   • Modern
   • Issues
3 Tuples
4 Trends
   • Coordination for Complex Systems
   • Coordination for Simulation
   • Coordination & Stochastic Systems
   • Full Dynamics
   • Core Mechanisms
   • Blending Metaphors
   • Predicting Complex Behaviours
   • Knowledge-oriented Coordination
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**

! nowadays, everybody knows about the essential role of **environment** in a MAS [Weyns et al., 2007]

? *do we also know how to design and engineer MAS environment?*
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

? *do we know how to embed stochastic behaviours in a MAS?*
Outline

1. Nature-inspired Coordination
2. Examples
3. Tuples
4. Trends
**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 Ancestor**
**LINDA is not a Nature-inspired Model**

So, *why LINDA?*

*Why tuple-based models?*
Why Tuple-based Models? I

**Expressiveness**

- **LINDA** is sort of a *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 \textit{permanent} coordination abstractions
- so, the \textit{coordination infrastructure} provides agents with tuple spaces as coordination services
  - \textit{coordination as a service (CaaS)} [Viroli and Omicini, 2006]
- they can be interpreted as \textit{coordination artefacts} shaping computational \textit{environment} [Omicini et al., 2004]
  - and used with different levels of awareness by both intelligent and “stupid” agents [Omicini, 2013a]
- as such, they can be exploited to support \textit{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, 2004a]
  - 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*
Just *some* is not enough

- capturing just *some* of the principles and mechanisms of natural systems does not ensure to capture their *essence*

- for instance, chemical coordination models such as Gamma and CHAM exploit the raw schema of computation as chemical reaction, but are *not* expressive enough to fully reproduce any non-trivial chemical system

- in fact, *e.g.*, even the simplest model for real chemical reactions requires a notion of *reaction rate*

- neither Gamma nor CHAM provide for such a notion, they are not expressive enough to fully match the behaviour of real chemical systems
Self-organising coordination [Viroli et al., 2009]

- most of the traditional coordination models feature abstractions enacting coordination laws that are typically *reactive*, (mostly) *deterministic*, and *global* as well
- in complex systems featuring self-* properties, instead, coordination patterns typically appear at the global level by *emergence*, from *probabilistic*, *time-dependent* coordination laws based on *local* criteria
- in particular, many coordination models either implicitly or explicitly recognise that full expressiveness requires addressing the issues of time dependency and stochasticity
**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

Time-aware ReSpecT

- ReSpecT [Omicini and Denti, 2001] generally addresses *time dependency* by capturing time events and supporting the definition and enforcement of *timed coordination policies* [Omicini et al., 2005]—so, ReSpecT-programmed tuple centres can work as time-dependent abstractions for MAS coordination [Omicini et al., 2007]
Enough?

No.

- in the overall, the above-mentioned models fail to capture all the essential features of nature-inspired coordination.
- this is why many novel research lines stretch existing tuple-based models to achieve the expressive power required to model and build MAS with a complexity comparable to natural systems [Omicini and Viroli, 2011]
Outline

1. Nature-inspired Coordination
2. Examples
3. Tuples
4. Trends
Outline

1. Nature-inspired Coordination
2. Examples
   - Early
   - Modern
   - Issues
3. Tuples
4. Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - Coordination & Stochastic Systems
   - Full Dynamics
   - Core Mechanisms
   - Blending Metaphors
   - Predicting Complex Behaviours
   - Knowledge-oriented Coordination
Complexity as a Multi-disciplinary Notion

Complex systems everywhere
- The notion of *complexity* is definitely a *multi-disciplinary* one, ranging from physics to biology, from economics to sociology and organisation sciences
- Systems that are said *complex* are both *natural* and *artificial* ones

Natural vs. artificial complex systems
- We *observe* and *model* complex *physical* systems
- We *design* and *build* complex *computational* systems

Question
- Which features do *all* complex systems *share* independently of their nature?
Complexity & Interaction

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

Laws of complexity

- Some “laws of complexity” exists that characterise any complex system, independently of its specific nature [Kauffman, 2003]
- The precise source of what all complex systems share is still unknown in essence

Interaction

- We argue that interaction – its nature, structure, dynamics – is the key to understand some fundamental properties of complex systems of any kind
Independence from interaction

- Some physical systems are described under the assumption of mutual independence among particles—that is, the behaviour of the particles is unaffected by their mutual interaction
  - e.g., ideal gas [Boltzmann, 1964]
- There, the probability distribution of the whole system is the product of those of each of its particles
- In computer science terms, the properties of the system can be compositionally derived by the properties of the individual components [Wegner, 1997]

→ Neither macroscopic sudden shift nor abrupt change for the system as a whole: technically, those systems have no phase transitions—of course, while the “independence from interaction” hypothesis holds
Interacting systems

- Introducing interaction among particles structurally *changes* the *macroscopic properties*, along with the *mathematical* ones.
- Interacting systems are systems where particles *do not behave independently* of each other.
- The probability distribution of an interacting system does not factorise anymore.
- In computer science terms, an interacting system is *non-compositional* [Wegner, 1997].
Interacting vs. non-interacting systems

- Only interacting systems can describe real cases beyond the idealised ones
  - e.g., they can explain phase transitions – like liquid-gas transition – and much more, such as collective emerging effects
- While a system made of independent parts can be represented by isolated single nodes, an interacting system is better described by nodes connected by lines or higher-dimensional objects
- From the point of view of information and communication theories, an ideal non-interacting gas is a system of non-communicating nodes, whereas an interacting system is made of nodes connected by channels
The case of magnetic particles

- The simplest standard prototype of an interacting system is the one made of magnetic particles.
- There, individual particles can behave according to a magnetic field which leaves their probabilistic independence undisturbed.
- At the same time, two magnetic particles interact with each other, and the strength of their interaction is a crucial tuning parameter to observe a phase transition.
  - If interaction is weak, the effect of a magnetic field is smooth on the system.
  - Instead, if the interaction is strong – in particular, higher than a threshold – even a negligible magnetic field can cause a powerful *cooperative effect* on the system.
Interaction is not enough

- Interaction is a necessary ingredient for complexity in statistical mechanics but definitely not a sufficient one.
- Complexity arises when the possible equilibrium states of a system grow very quickly with the number of particles, regardless of the simplicity of the laws governing each particle and their mutual interaction.
- Roughly speaking, complexity is much more related to size in number, rather than to complexity of the laws ruling interaction.

→ We do not need complex interaction to make interaction lead to complexity.
The key point in statistical mechanics is to relate the \textit{macroscopic} observables quantities—like pressure, temperature, etc.—to suitable \textit{averages} of \textit{microscopic} observables—like particle speed, kinetic energy, etc.

Based on the \textit{laws of large numbers}, the method works for those systems made of a \textit{large number} of particles / basic components.
Beyond the boundaries

- *Methods for complex systems* from statistical mechanics have expanded from physics to fields as diverse as biology [Kauffman, 1993], economics [Bouchaud and Potters, 2003, Mantegna and Stanley, 1999], and computer science itself [Mézard and Montanari, 2009, Nishimori, 2001]

- Recently, they have been applied to *social sciences* as well: there is evidence that the complex behaviour of many observed socio-economic systems can be approached with the *quantitative tools* from statistical mechanics
  - e.g., *Econophysics* for crisis events [Stanley, 2008]
Social systems as statistical mechanical systems

- A group of isolated individuals neither knowing nor communicating with each other is the typical example of a *compositional* social system.

- No sudden shifts are expected in this case at the collective level, unless it is caused by strong external exogenous causes.

- To obtain a *collective behaviour* displaying *endogenous* phenomena, the individual *agents* should meaningfully *interact* with each other.

- The foremost issue here is that the nature of the interaction determines the nature of the collective behaviour at the aggregate level.
  - e.g., a simple *imitative* interaction is capable to cause strong polarisation effects even in presence of extremely small external inputs.
Physical vs. computational systems

- Physical systems are to be observed, understood, and possibly modelled
  → For physical systems, the laws of interaction, and their role for complexity, are to be taken as given, to be possibly formalised mathematically by physicists

- Computational systems are to be designed and built
  → For computational systems, the laws of interaction have first to be defined through amenable abstractions and computational models by computer scientists, then exploited by computer engineers in order to build systems
Coordination media for ruling interaction

- Defining the abstractions for ruling the interaction space in computational systems basically means to define their *coordination model* [Gelernter and Carriero, 1992, Ciancarini, 1996, Ciancarini et al., 1999]

- **Global properties** of complex coordinated systems depending on interaction can be enforced through the *coordination model*, essentially based on its expressiveness [Zavattaro, 1998, Denti et al., 1998]
  - For instance, tuple-based coordination models have been shown to be expressive enough to support self-organising coordination patterns for nature-inspired distributed systems [Omicini, 2013b]
The role of coordination models

Coordination models could be exploited

- to *rule* the *interaction space*
- so as to *define* new sorts of *global*, macroscopic *properties* for *computational systems*, possibly inspired by physical ones
One should understand

- how to relate methods from statistical mechanics with coordination models
- whether notions such as *phase*, *phase transition*, or any other macroscopic system property, could be transferred from statistical mechanics to computer science
- what such notions would imply for computational systems
- whether new, original notions could apply to computational systems
- which sort of coordination model could support such notions
Socio-Technical Systems

Humans vs. software
- Nowadays, a particularly-relevant class of social systems is represented by socio-technical systems.
- In socio-technical systems:
  - active components are mainly represented by humans;
  - whereas interaction is almost-totally regulated by the software infrastructure;
  - where software agents often play a key role;
- This is the case, for instance, of social platforms like FaceBook [FaceBook, 2014] and LiquidFeedback [LiquidFeedback, 2014].
A twofold view of socio-technical systems

- The nature of socio-technical systems is twofold: they are both social systems and computational systems [Verhagen et al., 2013, Omicini, 2012]
- As *complex social systems*, their complex behaviour is in principle amenable of mathematical modelling and prediction through notions and tools from statistical mechanics
- As *complex computational systems*, they are designed and built around some (either implicit or explicit) notion of coordination, ruling the interaction within components of any sort—be them either software or human ones
Computational systems meet physical systems

- In socio-technical systems, macroscopic properties could be
  - described by exploiting the conceptual tools from physics
  - enforced by the coordination abstractions
- Socio-technical systems could exploit both
  - the notion of complexity by statistical mechanics, along with the mathematical tools for behaviour modelling and prediction, and
  - coordination models and languages to suitably shape the interaction space
Complex socio-technical systems could be envisioned
- whose implementation is based on suitable coordination models
- whose macroscopic properties can be modelled and predicted by means of mathematical tools from statistical physics
thus reconciling the scientist and the engineer views over systems
Outline

1. Nature-inspired Coordination
2. Examples
   - Early
   - Modern
   - Issues
3. Tuples
4. Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - Coordination & Stochastic Systems
   - Full Dynamics
   - Core Mechanisms
   - Blending Metaphors
   - Predicting Complex Behaviours
   - Knowledge-oriented Coordination
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
Case Study: Simulating Intracellular Signalling Pathways I

Intracellular signalling pathways

- Intracellular signalling involves several molecular processes along with a huge amount of signalling elements, including several kinds of proteins.
- Signal transduction pathways activated by G-proteins interact with one another to form a complex network that regulates diverse cellular components and controls a wide range of cellular processes [Neves et al., 2002].
- The Ras-regulated signal transduction pathways are a classical example of this kind of network [Downward, 2003].
Interaction issues in intracellular signalling pathways

- To model intracellular signalling systems, complex interaction that governs their behaviour should be first of all considered and understood.
- Though determining the kinetic equations of the biochemistry involved in vital functions is important, managing interactions for the cell to make the correct physiological decisions is even more so.
- Simulation of intracellular signalling pathways could be framed as mostly a coordination issue.
Case Study: Simulating Intracellular Signalling Pathways III

Biochemical coordination

- **Biochemical tuple spaces** [Viroli and Casadei, 2009] are the core of a model for self-organising coordination (BTS-SOC).
- A biochemical tuple space is a tuple space working as a compartment where biochemical reactions take place.
- Tuples in BTS-SOC are associated with an activity/pertinency value, resembling chemical concentration, and allowing chemical reactants to be represented as tuples.
- Biochemical laws are represented as **coordination laws** by the coordination abstraction, evolving tuple concentration over time according to a rate in the same way as chemical substances into a solution.
- Also, BTS-SOC laws allow for tuple diffusion, making it possible for products to cross compartment boundaries as a result of biochemical reactions.
### Case Study: Simulating Intracellular Signalling Pathways IV

Mapping cellular components and structures involved in intracellular signalling onto BTS-SOC abstractions [González Pérez et al., 2013]

<table>
<thead>
<tr>
<th>Cellular components and structures involved in intracellular signalling</th>
<th>Computational abstractions of the BTS-SOC model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extracellular milieu and intracellular compartments (<em>i.e.</em>, membrane, juxtamembrane region, cytosol, nucleus)</td>
<td>Tuple centres</td>
</tr>
<tr>
<td>Signalling components (<em>i.e.</em>, membrane receptors, proteins, enzymes and genes)</td>
<td>Chemical reactions sets</td>
</tr>
<tr>
<td>Signalling molecules (<em>i.e.</em>, first and secondary messengers), activation and deactivation signals</td>
<td>Reactants and concentrations recorded as tuples in the tuple centre</td>
</tr>
</tbody>
</table>
Case Study: Simulating Intracellular Signalling Pathways V

Trends
Coordination for Simulation

Tuple centre (extracellular milieu)

Tuple centre (membrane)
Set of chemical reactions
Set of chemical reactions
Set of chemical reactions

Tuple centre (juxta membrane region)
Set of chemical reactions
Set of chemical reactions
Set of chemical reactions

Tuple centre (cytosol)
Set of chemical reactions
Set of chemical reactions
Set of chemical reactions

Tuple centre (nucleus)
Set of chemical reactions
Set of chemical reactions
Set of chemical reactions

Coordination tool: ReSpecT chemical engine
Coordination tool: ReSpecT chemical engine
Coordination tool: ReSpecT chemical engine
Coordination tool: ReSpecT chemical engine

The results of trans-disciplinary research efforts may appear quite obvious, once they are seen a posteriori.

Just above, a nature-inspired model developed in computational terms (biochemical tuple spaces) is exploited as a computational support to the simulation of a natural system (intracellular signalling pathways).

In other terms, from the natural world to the computational one, and back—and it works, as one might expect.
However, one should also understand that trans-disciplinary research succeeds when each translation of findings between the different fields involved actually enriches the associated concepts and techniques.

- Above, the BTS-SOC approach features the properties deriving from its biochemical inspiration along with those of tuple-based coordination for complex computation systems.
- When brought back to the ‘natural domain’ as a tool for biochemical simulation, BTS-SOC fits well for its natural inspiration, but its good performance in terms of expressive capabilities and computational efficiency also depends on its tuple-based structure.
So, while natural inspiration does not per se ensure the appropriateness of a computational approach to natural system simulation, it may in principle provide a sound grounding for the simulation of natural systems.

- Biochemical inspiration of the BTS-SOC model seems to couple well with the properties of tuple-based coordination.
- BTS-SOC turns out to be a suitable framework for the simulation of biochemical systems.
Outline

1. Nature-inspired Coordination
2. Examples
   - Early
   - Modern
   - Issues
3. Tuples
4. Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - **Coordination & Stochastic Systems**
   - Full Dynamics
   - Core Mechanisms
   - Blending Metaphors
   - Predicting Complex Behaviours
   - Knowledge-oriented Coordination
Probability

- *Probability* measures how likely some event will occur.
- At its core, probability provides a mathematical framework to describe *casual* events.
- ... where casual essentially means *non-deterministic*.
- By definition, probability deals with *single occurrences* of events.
- From a scientific viewpoint, a probabilistic description *per se* has *no predictive value*: it cannot really predict the precise outcome of a phenomenon.
- In any case, probability provides an *a priori* model for non-deterministic phenomena.
Statistics & Probability

- **Statistics** describes / analyses / interprets phenomena starting from the data available about them.
- Whenever a phenomenon has no *a priori* mathematical model (at least, not yet), statistics is concerned with getting one *a posteriori* from the available data.
- Accordingly, statistics is concerned with *several occurrences* of (non-deterministic) events.
- Probability typically provide the mathematical tools to explain data & build a model.
Stochastic Systems

- *Stochastic systems* are non-deterministic systems
- A stochastic system is one whose states are determined probabilistically
- More generally, any phenomenon requiring probability for its description is (at least in part) stochastically by definition
- Roughly speaking, a probabilistic model for a stochastic system provides a *predictive framework* for a non-deterministic system
  → We cannot predict the single occurrence of a non-deterministic event, but we can predict the overall outcome of repeated occurrences of a non-deterministic event
Non-determinism, Coordination & Stochastic Behaviour

- Autonomous systems such as *adaptive* and *self-* ones are *stochastic systems* at their very heart.
- Accordingly, a foremost feature of computational models for adaptive and self-* systems is *non-determinism*.
- In order to obtain stochastic behaviours of computational systems, suitable mechanisms for non-determinism should be provided.
- Since most of the complexity featured by adaptive and self-* systems depends on the interaction among components, *coordination models* should feature *non-deterministic coordination mechanisms* for stochastic behaviour.
Issues

- Devising out some *basic mechanisms* for stochastic coordination
- Finding a *minimal* set of primitives for most (all) of the most relevant stochastic systems
- Showing how such mechanisms could be embedded as *tuple-based* co-ordination primitives, in order to address the general need of complex computational system engineering
- Defining their *formal semantics* and implementing them as TuCSoN primitives
Don’t Care Non-determinism in Tuple-based Models

- **LINDA** features *don't know* non-determinism handled with a *don't care* approach:
  
  *don't know* which tuple among the matching ones is retrieved by a getter operation (*in*, *rd*) can be neither specified nor predicted
  
  *don't care* nonetheless, the coordinated system is designed so as to keep on working whichever is the matching tuple returned

- Instead, adaptive and self-organising systems require stochastic behaviours like "*most of the time do this*", "*sometimes do that*"

- Possibly with some quantitative specification of "*most of the time*" and "*sometimes*"

→ As it is, non-determinism in tuple-based models does not fit the need of stochastic behaviour specification
**LINDA “Local” Nature – In Time & Space**

**No context** — In a single getter operation, only a *local*, point-wise property affects tuple retrieval: that is, the conformance of a tuple to the template, independently of the *spatial* context.

- In fact, standard getter primitives return a matching tuple independently of the other tuples currently in the same space—so, they are “*context unaware*”

**No history** — Furthermore, in a sequence of getter operations, don’t know non-determinism makes any prediction of the overall behaviour impossible. Again, then, only a point-wise property can be ensured even in *time*.

- Sequences of standard getter operations present no meaningful distribution over time.
**Linda:** How to Roll a Dice?

- We define tuple space dice
- We represent a six-face dice as a collection of six tuples: face(1), ..., face(6)
- We roll a dice by rd-ing a face/1 tuple from dice:
  
  \[ \text{dice} \ ? \ rd(\text{face}(X)) \]

! We do *not* obtain the overall (stochastic) behaviour of a dice: for instance, it may reasonably happen that rolling the dice \(10^9\) times *always* results in \(X / 1\)—that is, we get “1” \(10^9\) times in a row.
**uLINDA: Probabilistic Non-determinism**

- We define **uniform coordination primitives** \((u_{in}, u_{rd})\) – first mentioned in [Gardelli et al., 2007] – as the **specialisation** of LINDA getter primitives featuring **probabilistic non-determinism** instead of don’t know non-determinism.
- We call the new model **uLINDA** [Mariani and Omicini, 2013c].
- Uniform primitives allow programmers to both specify and **(statistically) predict** the probability to retrieve one specific tuple among a bag of matching tuples.
- Uniform primitives are the **“basic mechanisms enabling self-organising coordination”**—that is, a minimal set of constructs able (**alone**) to impact the observable properties of a coordinated system.
**uLINDA: “Global” Nature**

**Situation & prediction**

Uniform primitives replace don’t know non-determinism with *probabilistic non-determinism* to

- **situ**ate a primitive invocation in space
  - uniform getter primitives return matching tuples based on the other tuples in the space—so, their behaviour is *context-aware*.

- **predict** its behaviour in time
  - sequences of uniform getter operations tend to globally exhibit a *uniform distribution* over time.
uLINDA: How to Roll a Dice?

- Again, we define tuple space dice
- Again, we represent a six-face dice as a collection of six tuples: face(1), ..., face(6)
- We roll a dice by urd-ing a face/1 tuple from dice:
  \[
  \text{dice ? urd(face(X))}
  \]

! Now, we do obtain the overall (stochastic) behaviour of a dice:

- context — at every roll, the six faces of the dice X / 1, ..., X / 6 have the same probability P = 1/6 to be selected
- history — in the overall, repeating several times a roll, the six faces will tend to converge towards a uniform distribution
Informal Semantics

Operationally, uniform primitives behave as follows:

1. When executed, a uniform primitive takes a *snapshot* of the tuple space, “freezing” its state at a certain point in time—and space, being a single tuple space the target of basic *LINDA* primitives.

2. The snapshot is then exploited to assign a probabilistic value $p_i \in [0, 1]$ to any tuple $t_i \in \{1..n\}$ in the space—where $n$ is the total number of tuples in the space.

3. There, non-matching tuples have value $p = 0$, matching tuples have value $p = 1/m$ (where $m \leq n$ is the number of matching tuples), and the overall sum of probability values is $\sum_{i=1..n} p_i = 1$.

4. The choice of the matching tuple to be returned is then statistically based on the computed probabilistic values.
In order to define the semantics of (getter) uniform primitives, we rely upon a simplified version of the process-algebraic framework in [Bravetti, 2008], in particular the $\uparrow$ operator, dropping multi-level priority probabilities.

\[ \text{uin semantics} \]

**[Synch-C]** \( \text{uin}_T.P \mid \langle t_1, .., t_n \rangle \xrightarrow{T} \text{uin}_T.P \mid \langle t_1, .., t_n \rangle \uparrow \{(t_1, v_1), .., (t_n, v_n)\} \)

**[Close-C]** \( \text{uin}_T.P \mid \langle t_1, .., t_n \rangle \uparrow \{(t_1, v_1), .., (t_n, v_n)\} \)

\[ \leadsto \]

\( \text{uin}_T.P \mid \langle t_1, .., t_n \rangle \uparrow \{(t_1, p_1), .., (t_n, p_n)\} \)

**[Exec-C]** \( \text{uin}_T.P \mid \langle t_1, .., t_n \rangle \uparrow \{.., (t_j, p_j), ..\} \xrightarrow{t_j} P[t_j / T] \mid \langle t_1, .., t_n \rangle \backslash t_j \)
As for standard Linda getter primitives, the only difference between uniform reading (urd) and uniform consumption (uin) is the non-destructive semantics of the reading primitive—transition Exec-R.

urd semantics

\[
\begin{align*}
\text{[Synch-C]} & \quad \text{uin}_T. P \mid \langle t_1, \ldots, t_n \rangle \xrightarrow{T} \text{uin}_T. P \mid \langle t_1, \ldots, t_n \rangle \uparrow \{ (t_1, v_1), \ldots, (t_n, v_n) \} \\
\text{[Close-C]} & \quad \text{uin}_T. P \mid \langle t_1, \ldots, t_n \rangle \uparrow \{ (t_1, v_1), \ldots, (t_n, v_n) \} \\
& \quad \iff \\
& \quad \text{uin}_T. P \mid \langle t_1, \ldots, t_n \rangle \uparrow \{ (t_1, p_1), \ldots, (t_n, p_n) \} \\
\text{[Exec-R]} & \quad \text{uin}_T. P \mid \langle t_1, \ldots, t_n \rangle \uparrow \{ \ldots, (t_j, p_j), \ldots \} \xrightarrow{t_j} p_j \ P[t_j/T] \mid \langle t_1, \ldots, t_n \rangle
\end{align*}
\]
Expressiveness: $uLINDA$ vs $LINDA$

In [Bravetti et al., 2005], authors demonstrate that $LINDA$-based languages cannot implement probabilistic models.

**PME proof**

The gain in expressiveness brought by $uLINDA$ is formally proven in [Mariani and Omicini, 2013a], where uniform primitives are shown to be strictly more expressive than standard $LINDA$ primitives according to probabilistic modular embedding (PME) [Mariani and Omicini, 2013b].

In particular

$$uLINDA \succeq_p LINDA \land LINDA \not\succeq_p uLINDA \implies uLINDA \not\equiv_o LINDA$$

where

- $\succeq_p$ stands for “probabilistically embeds”
- $\equiv_o$ means “(PME) observational equivalence”
Outline

1. Nature-inspired Coordination
2. Examples
   - Early
   - Modern
   - Issues
3. Tuples
4. Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - Coordination & Stochastic Systems
   - Full Dynamics
   - Core Mechanisms
   - Blending Metaphors
   - Predicting Complex Behaviours
   - Knowledge-oriented Coordination
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, Banâtre 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
Outline

1 Nature-inspired Coordination
2 Examples
   - Early
   - Modern
   - Issues
3 Tuples
4 Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - Coordination & Stochastic Systems
   - Full Dynamics
   - Core Mechanisms
   - Blending Metaphors
   - Predicting Complex Behaviours
   - Knowledge-oriented Coordination
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.
Outline

1. Nature-inspired Coordination
2. Examples
   - Early
   - Modern
   - Issues
3. Tuples
4. Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - Coordination & Stochastic Systems
   - Full Dynamics
   - Core Mechanisms
   - **Blending Metaphors**
   - Predicting Complex Behaviours
   - Knowledge-oriented 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, 2004b] 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.
Outline

1. Nature-inspired Coordination
2. Examples
   - Early
   - Modern
   - Issues
3. Tuples
4. Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - Coordination & Stochastic Systems
   - Full Dynamics
   - Core Mechanisms
   - Blending Metaphors
   - Predicting Complex Behaviours
   - Knowledge-oriented Coordination
Predicting Complex Behaviours

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
Outline

1. Nature-inspired Coordination
2. Examples
   - Early
   - Modern
   - Issues
3. Tuples
4. Trends
   - Coordination for Complex Systems
   - Coordination for Simulation
   - Coordination & Stochastic Systems
   - Full Dynamics
   - Core Mechanisms
   - Blending Metaphors
   - Predicting Complex Behaviours
   - Knowledge-oriented Coordination

Andrea Omicini (DISI, Univ. Bologna)
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
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
MoK (Molecules of Knowledge) [Mariani and Omicini, 2012a]

- Molecules of Knowledge is 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]
History and evolution

- starting from early chemical and stigmergic approaches, nature-inspired models of coordination evolved to become the potential core of complex MAS—such as pervasive, knowledge-intensive, and self-* MAS
- in this talk we briefly surveyed their history, devised their main issues, and pointed out the most promising trends
- focusing in particular on tuple-based coordination models, and adopted a systemic view over MAS
nature-inspired models of coordination already have a long history behind them

and apparently a huge potential for development still to be explored

to provide core abstractions and technologies for the engineering of complex MAS
1 Nature-inspired Coordination

2 Examples

3 Tuples

4 Trends


Bibliography II

*Lectures on Gas Theory.*
University of California Press.


Expressing priorities and external probabilities in process algebra via mixed open/closed systems. 
Bibliography III


FaceBook (2014).
Home page.

Triple-space computing: Semantic web services based on persistent publication of information.

Coordination languages and their significance.

A biochemically-inspired coordination-based model for simulating intracellular signalling pathways.
Special Issue: Agent-based Modeling and Simulation.

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.


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*,

Nature-inspired coordination for complex distributed systems.
In Fortino, G., Bădică, C., Malgeri, M., and Unland, R., editors,
Invited paper.

The multidisciplinary patterns of interaction from sciences to Computer Science. 

Timed environment for Web agents. 
Web Intelligence and Agent Systems, 5(2):161–175.


1st International Workshop on Self-Organizing Architectures (SOAR 2009), Cambridge, UK, 14-17 September 2009, Revised Selected and Invited Papers.


Nature-Inspired Coordination & Self-Organisation

Autonomous Systems
Sistemi Autonomi

Andrea Omicini
andrea.omicini@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
ALMA MATER STUDIORUM – Università di Bologna

Academic Year 2014/2015