

# Blending Event-Based and Multi-Agent Systems around Coordination Abstractions

Andrea Omicini, Giancarlo Fortino, Stefano Mariani

ALMA MATER STUDIORUM—Università di Bologna, Italy  
Università della Calabria, Italy

COORDINATION 2015  
Grenoble, France  
3 June 2015



- 1 Premises
- 2 Multi-Agent Systems as Coordinated Systems
- 3 Event-based Systems as Coordinated Systems
- 4 EBS & MAS: Towards a Unified Framework
- 5 Conclusion

# Outline

- 1 Premises
- 2 Multi-Agent Systems as Coordinated Systems
- 3 Event-based Systems as Coordinated Systems
- 4 EBS & MAS: Towards a Unified Framework
- 5 Conclusion



# Paradigms for Complex Systems Engineering

- **Multi-agent systems** (MAS) provide the most viable abstractions to model and engineer complex software systems [Jen00, Jen01]
- **Event-based systems** (EBS) are prevalent for large-scale distributed applications [FMG02], addressing accidental complexity due to, e.g., distributed interaction and large-scale concurrency [Bro87, MJGM13]

## State-of-art

MAS and EBS stand as the two most promising paradigms for *complex systems modelling and engineering*.

# The Idea

- Can agent-based and event-based models and technologies *coexist* without harming the conceptual integrity of systems?
- Is there some *common conceptual foundation* that could be exploited as a grounding for integrating agent-based and event-based abstractions, languages, technologies, and methods, in a coherent and effective way?
- Which *benefits* could such integration bring to both MAS and EBS engineers?

# The Claim

- The **interaction** dimension of computation is essential in both MAS and EBS

## Integration means

**Coordination** abstractions, models and technologies could play an essential role in MAS and EBS integration...

- ...*preserving* conceptual integrity of systems.
- ✓ EBS engineers gain *increased expressiveness* and better abstractions for the modelling & engineering of their EBS
- ✓ MAS engineers gain a solid technological foundation and an *industry-recognised* architectural style to model & engineer MAS

# The Approach

- 1 Recognise the *essential* abstractions of both paradigms as well as their mutual interpretation, in particular w.r.t. coordination issues
- 2 Devise out three steps towards a **unified conceptual framework** for EBS and MAS, laying the *foundation* of a coherent discipline for the modelling and engineering of complex software systems
- 3 Recognise/define the fundamental role of **coordination abstractions** in such framework

# Outline

- 1 Premises
- 2 Multi-Agent Systems as Coordinated Systems**
- 3 Event-based Systems as Coordinated Systems
- 4 EBS & MAS: Towards a Unified Framework
- 5 Conclusion





# Main abstractions of MAS

- MAS can be modelled around three first-class abstractions [WOO07]:
  - agents
  - societies
  - environment

# Agents

- **Agents** are computational entities whose defining feature is *autonomy* [Woo02, ORV08]
- ⇒ MAS are collections of *social* autonomous agents, working together toward a coherent system goal

## Goals & activities

- Agents model *activities* within a MAS, as a composition of observable actions...
  - ...along with the motivations behind them—namely, the *goals* driving the agents' course of actions [Cas12].

# Societies

- **Societies** represent the ensembles where the collective behaviours of the MAS are *regulated* towards the achievement of the overall system goals. . .
    - . . . by coordinating their mutual *dependencies* [MC94]—as a form of interaction
- ⇒ **Coordination models** [GC92] are the most suitable abstraction to harness dependencies [MC94], thus *interaction complexity* [COZ00]

## Dependencies & coordination

- Agent societies can then be built around the abstractions provided by coordination models [COZ00]:
  - *coordination media* [Cia96], in the general case of distributed systems
  - *coordination artefacts* [ORV<sup>+</sup>04], in the case of MAS

# Environment

- **Environment** captures the *unpredictability* of the MAS context...
  - ...by modelling the *external resources and properties* that are relevant for the MAS...
  - ...along with their *change over time* [WOO07]
- *Situated action* is the realisation that coordinated, social action arises from strict interaction with the environment, rather than from rational practical reasoning [Suc87]

## Situated action & MAS situatedness

- The need for *situatedness* in MAS demands for being sensitive to *environment change* [FM96]
- ⇒ This basically means *dependency*, again

# MAS as Coordinated Systems I

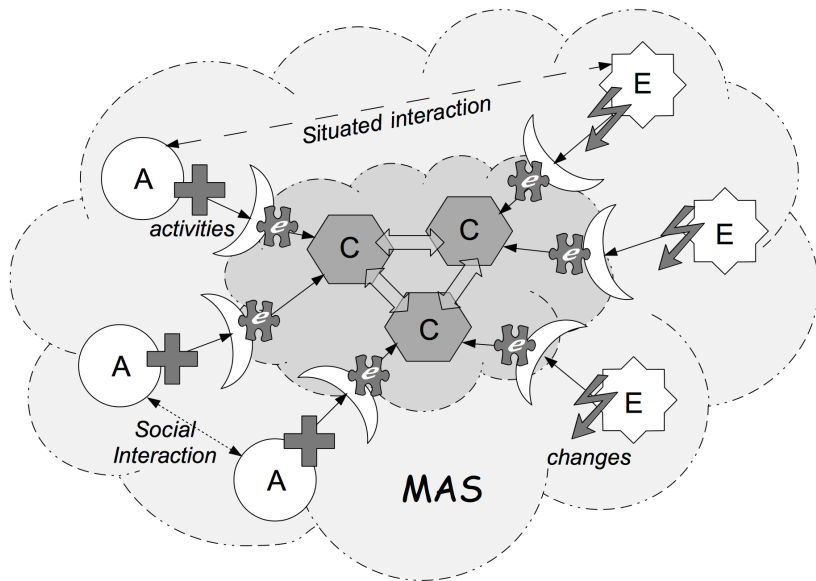
Summing up:

- things happen in a MAS because of either agent activity or environment change
- complexity arises from both *social* and *situated* interaction

## Role of coordination

- Coordination can be used to deal with both *social* and *situated* dependencies in a uniform way
- **Coordination artefacts** could be exploited to handle both social and situated interaction [MO15]

# MAS as Coordinated Systems II



# Outline

- 1 Premises
- 2 Multi-Agent Systems as Coordinated Systems
- 3 Event-based Systems as Coordinated Systems**
- 4 EBS & MAS: Towards a Unified Framework
- 5 Conclusion



# Terminology for Event-Based Systems

## Definition

According to [FMG02], an **event-based systems** is

*a system in which the integrated components communicate by generating and receiving **event notifications***

- An **event** is the occurrence of a happening relevant for the system, e.g. a state change in some component or system property
- A **notification** is the reification of an event within the system, and provides for its description



# Event-Based Architectures

- Components in event-based systems act as either **producers** or **consumers** of notifications
    - Producers **publish** notifications, and provide an **output interface** for subscription
    - Consumers **subscribe** to notifications, as specified by producers' interfaces
  - Notifications are distributed by the **event notification service**—implemented by the **event bus**
- ⇒ Expressiveness of subscriptions depends on the data model, filter language, and subscription interfaces used

# Complex Event Processing

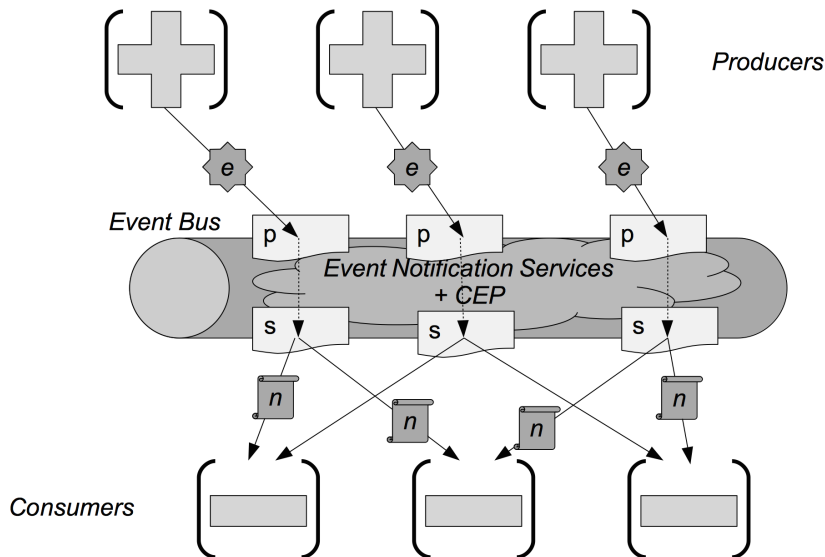
- Articulated application scenarios raise the need for **Complex Event Processing** (CEP) [Luc02]—e.g., Business Activity Monitoring, Sensor Networks, and Market data analysis
  - event aggregation
  - event transformation
  - event correlation (such as causality, timing, membership, hierarchy)
  - event abstraction
  - ...
- CEP is usually responsibility of the event notification service itself

# EBS as Coordinated Systems I

*EBS are coordinated systems* in their essence:

- coordination is event-based [MJGM13], and follows a publication/subscription model—specific to the EBS
- the event bus is the system *coordinator*, encapsulating and automating most of the coordination activities, e.g. CEP-related ones
- producer and consumer components are the subjects of the coordination process—the *coordinables*, according to [Cia96]
- ! control of the logic of program execution is somehow *inverted* [HO06]

# EBS as Coordinated Systems II



# Outline

- 1 Premises
- 2 Multi-Agent Systems as Coordinated Systems
- 3 Event-based Systems as Coordinated Systems
- 4 EBS & MAS: Towards a Unified Framework**
- 5 Conclusion



# Outline

- 1 Premises
- 2 Multi-Agent Systems as Coordinated Systems
- 3 Event-based Systems as Coordinated Systems
- 4 EBS & MAS: Towards a Unified Framework
  - **Event Sources**
  - Boundary Artefacts
  - Event-based Coordination
- 5 Conclusion



# Event-based View of MAS

- *Agents* are the *designed* source of events, autonomously driving control towards their own *goals*, and producing *internal events* through actions
- *Environment* is the source of *external events*, whose dynamics is in principle *unpredictable*, through dedicated abstractions capturing *heterogeneity* of the environment

## MAS as EBS

- Agents and environmental abstractions are both producers and consumers of events (*prosumers*)—internal and external, respectively.

# Agent-based View of EBS I

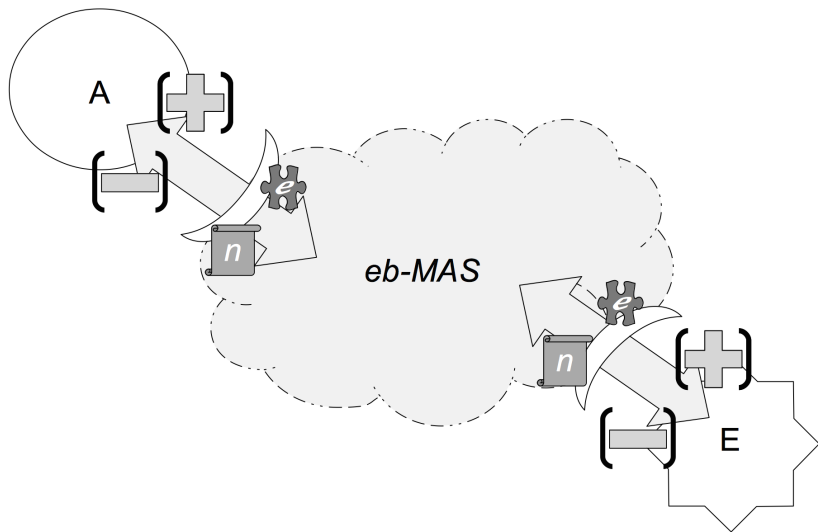
- **Producers** of events can be an **agent**, undertaking *social actions*, or the **environment**, due to its unpredictable *dynamics* or due to agents *situated actions*
- **Consumers** too can be an **agent**, whose actions depend on other agents' (social) actions, or the environment, who is reactive to agents' (situated) actions

## EBS as MAS

- Producers and consumers are not mutually exclusive components, but *roles* played simultaneously by agents and environmental abstractions according to the (dynamics of) EBS at hand.



# Agent-based View of EBS II



# Benefits

## To EBS engineers

- Higher **expressiveness**
  - ⇒ producers and consumers can be interpreted according to a novel abstraction layer, that is agents-environment, modelling events sources according to their **causes**—e.g., goals behind activities

## To MAS engineers

- Improved **conceptual integrity**
  - ⇒ both agents and the environment are first-class event-based abstractions, thus causes of events in a MAS are modelled in a uniform way as **event prosumers**

# Outline

- 1 Premises
- 2 Multi-Agent Systems as Coordinated Systems
- 3 Event-based Systems as Coordinated Systems
- 4 EBS & MAS: Towards a Unified Framework
  - Event Sources
  - **Boundary Artefacts**
  - Event-based Coordination
- 5 Conclusion

# Dealing with Heterogeneity

- Agents and environment are *heterogeneous* sources of events:
    - physical acts, communications, etc., for agents
    - actuation, perception, etc., for the environment
- ⇒ Heterogeneity mandates for a *general event model*, capable of capturing any sort of event – along with its properties – in an *expressive yet uniform* way

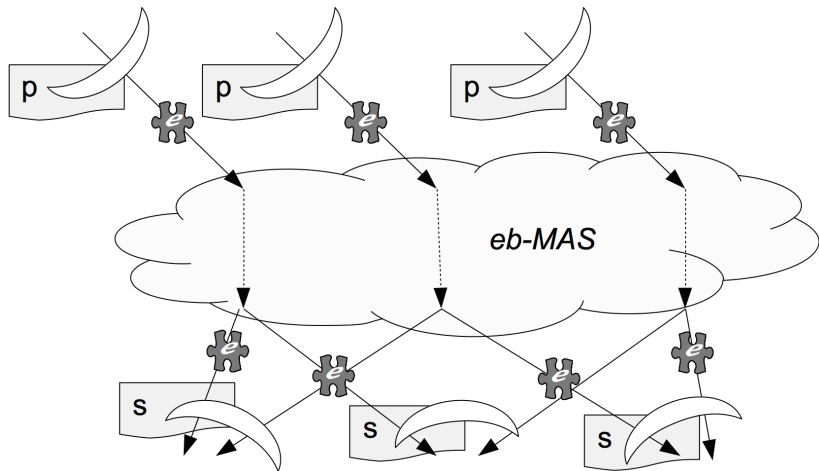
## Mediating architectural abstraction

*Boundary artefacts* [MO15] allow all sorts of happenings to be suitably *translated*, according to the common event model, and *mediated* into a *uniform system of notifications*.

# An Architectural Pattern I

- In MAS:
  - for agents, *Agent Communication Contexts* [DSPST06], *Agent Coordination Contexts* [Omi02]
  - for the environment, environment machinery in *Jason* [BHW07], *transducers* for probes [MO14] in ReSpecT
- In EBS:
  - *event mediators* (or, *correlators*) in the Cambridge Event Architecture [BMB<sup>+</sup>00]—as implemented by, e.g., Hermes event-based middleware [PB02], OASIS open service architecture [BM02]

# An Architectural Pattern II



# Benefits

## To EBS engineers

- Enable *autonomy*
  - ⇒ as they rule interaction within an EBS as *constrainers*, rather than *commanders* [Weg96], thus components can be autonomous—in line with the agent abstraction

## To MAS engineers

- Enable *uniformity* in coordination
  - ⇒ as they *mediate* between sources of events and events notifications, thus heterogeneity of interactions (social vs. situated) maps to a *uniform* handling of event dependencies—that is, event-based coordination

# Outline

- 1 Premises
- 2 Multi-Agent Systems as Coordinated Systems
- 3 Event-based Systems as Coordinated Systems
- 4 EBS & MAS: Towards a Unified Framework
  - Event Sources
  - Boundary Artefacts
  - **Event-based Coordination**
- 5 Conclusion





# Dealing with Dependencies

- A **coordination model** provides a framework in which the dependencies between (pro-/re-)active and independent entities can be expressed [Cia96]
  - **coordinables** are the entities whose mutual interaction is ruled by the model
  - **coordination media** are the abstractions enabling and ruling interaction
  - **coordination laws** are the rules governing the observable behaviour of coordinables, the behaviour of the coordination media, as well as their interactions

# Event-Based Coordination I

- A number of
  - event-based *middleware*—e.g., JEDI [CDNF01]
  - event-based *coordination models* [PA98, VOR02]
  - event-based *coordination technologies*—e.g., JavaSpaces [FHA99]
  - event-based *coordination formalisms* [Omi99, VR02]
- in literature witness
  - the role of event-based middleware in the engineering of complex distributed systems
  - the event-based nature of the most relevant coordination models, including tuple-based ones [Omi99]

# Event-Based Coordination II

The last step in the integration of MAS and EBS is the recognition that

- **coordination media** [Cia96] are the abstractions in charge of coordinating multiple *event flows* [ORZ06]...
- ...according to their *mutual dependencies*—events publication/subscription in EBS, social/situated interaction in MAS

# Event-Based Coordination III

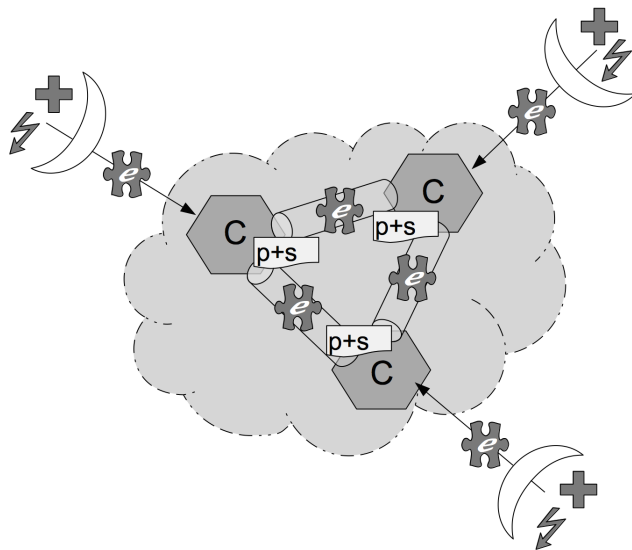
## MAS as EBS

- The *coordination media* should be built out of event-based *coordination artefacts* [MO15], the architectural abstractions aimed at governing and ruling coordination in MAS

## EBS as MAS

- The *coordination media* should work as the core of an *event-based architecture*, where event-related services should be designed around coordination abstractions [VO06]

# Event-Based Coordination IV



# Benefits

## To EBS engineers

- Improved **programming paradigm**
  - ⇒ coordination artefacts counterfeit inversion of control, by encapsulating the logic of coordination of multiple flows of events—originating from *autonomous* entities

## To MAS engineers

- Improved **non-functional properties**
  - ⇒ event-based architectures deployed in industry, once interpreted according to MAS abstractions, provide a suitable middleware on top of which to design high-performance, dependable, large-scale (event-based) coordination services

# Outline

- 1 Premises
- 2 Multi-Agent Systems as Coordinated Systems
- 3 Event-based Systems as Coordinated Systems
- 4 EBS & MAS: Towards a Unified Framework
- 5 Conclusion**



# Summing Up I

## A unified conceptual framework for MAS and EBS...

- ... is possible
- ... could work as the *foundation* of a principled discipline for the engineering of complex software systems
- ... by promoting a coherent *integration* of agent-based and event-based abstractions, languages, technologies, and methods



# Summing Up II

## In this position paper

- We suggest that a fundamental role in such a conceptual framework could be played by *coordination* models and technologies. . .
- . . . with the focus on *coordination artefacts* working as both event-based and agent-based abstractions
- Event-based coordination models and middleware could provide the conceptual and technical grounding for *complex system engineering*, integrating agent-based and event-based tools and techniques without harming conceptual integrity

# Summing Up III

## Steps

Three conceptual steps towards the unified framework

- ① Agents and environment are interpreted as the only sources of events
- ② Boundary artefacts as the architectural abstractions mediating between event sources and the system, promoting a uniform event model
- ③ Coordination media as the architectural abstractions for coordinating multiple event flows

# References I



Rafael H. Bordini, Jomi F. Hübner, and Michael J. Wooldridge.  
*Programming Multi-Agent Systems in AgentSpeak using Jason*.  
John Wiley & Sons, Ltd, October 2007.



Jean Bacon and Ken Moody.  
Toward open, secure, widely distributed services.  
*Communications of the ACM*, 45(6):59–64, June 2002.



Jean Bacon, Ken Moody, John Bates, Richard Heyton, Chaoying Ma, Andrew McNeil,  
Oliver Seidel, and Mark Spiteri.  
Generic support for distributed applications.  
*Computer*, 33(3):68–76, March 2000.



Frederick P. Brooks.  
No Silver Bullet Essence and Accidents of Software Engineering.  
*Computer*, 20(4):10–19, April 1987.



Cristiano Castelfranchi.  
Goals, the true center of cognition.  
In Fabio Paglieri, Luca Tummolini, Rino Falcone, and Maria Miceli, editors, *The Goals of Cognition. Essays in Honor of Cristiano Castelfranchi*, volume 20 of *Tributes*, chapter 41,  
pages 837–882. College Publications, London, December 2012.

# References II



Gianpaolo Cugola, Elisabetta Di Nitto, and Alfonso Fuggetta.

The JEDI event-based infrastructure and its application to the development of the OPSS WFMS.

*IEEE Transactions on Software Engineering*, 27(9):827–850, September 2001.



Paolo Ciancarini.

Coordination models and languages as software integrators.

*ACM Computing Surveys*, 28(2):300–302, June 1996.



Paolo Ciancarini, Andrea Omicini, and Franco Zambonelli.

Multiagent system engineering: The coordination viewpoint.

In Nicholas R. Jennings and Yves Lespérance, editors, *Intelligent Agents VI. Agent Theories, Architectures, and Languages*, volume 1757 of *LNAI*, pages 250–259. Springer, 2000.

6th International Workshop (ATAL'99), Orlando, FL, USA, 15–17 July 1999. Proceedings.



Antonella Di Stefano, Giuseppe Pappalardo, Corrado Santoro, and Emiliano Tramontana.

The transparent implementation of agent communication contexts.

*Concurrency and Computation: Practice and Experience*, 18(4):387–407, 2006.

# References III



Eric Freeman, Susanne Hupfer, and Ken Arnold.

*JavaSpaces Principles, Patterns, and Practice: Principles, Patterns and Practices.*  
The Jini Technology Series. Addison-Wesley Longman, June 1999.



Jacques Ferber and Jean-Pierre Müller.

Influences and reaction: A model of situated multiagent systems.

In Mario Tokoro, editor, *2nd International Conference on Multi-Agent Systems (ICMAS-96)*, pages 72–79, Tokio, Japan, December 1996. AAAI Press.



Ludger Fiege, Gero Mühl, and Felix C. Gärtner.

Modular event-based systems.

*The Knowledge Engineering Review*, 17(4):359–388, December 2002.



David Gelernter and Nicholas Carriero.

Coordination languages and their significance.

*Communications of the ACM*, 35(2):97–107, 1992.

# References IV



Philipp Haller and Martin Odersky.

Event-based programming without inversion of control.

In David E. Lightfoot and Clemens Szyperski, editors, *Modular Programming Languages. 7th Joint Modular Languages Conference, JMLC 2006 Oxford, UK, September 13-15, 2006 Proceedings*, number 1 in Lecture Notes in Computer Science, pages 4–22, Oxford, UK, September 2006. Springer Berlin Heidelberg.



Nicholas R. Jennings.

On agent-based software engineering.

*Artificial Intelligence*, 117(2):277–296, March 2000.



Nicholas R. Jennings.

An agent-based approach for building complex software systems.

*Communications of the ACM*, 44(4):35–41, April 2001.



David C. Luckham.

*The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*.

Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 2002.

# References V



Thomas W. Malone and Kevin Crowston.  
The interdisciplinary study of coordination.  
*ACM Computing Surveys*, 26(1):87–119, 1994.



Aleksandar Milicevic, Daniel Jackson, Milos Gligoric, and Darko Marinov.  
Model-based, event-driven programming paradigm for interactive Web applications.  
In *2013 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software (Onward!'13)*, pages 17–36, New York, NY, USA, October 2013. ACM Press.



Stefano Mariani and Andrea Omicini.  
Coordination in situated systems: Engineering mas environment in TuCSon.  
In Giancarlo Fortino, Giuseppe Di Fatta, Wenfeng Li, Sergio Ochoa, Alfredo Cuzzocrea, and Mukaddim Pathan, editors, *Internet and Distributed Computing Systems*, volume 8729 of *Lecture Notes in Computer Science*, pages 99–110. Springer International Publishing, September 2014.  
7th International Conference on Internet and Distributed Computing Systems (IDCS 2014), Calabria, Italy, 22-24 September 2014, Proceedings.



Stefano Mariani and Andrea Omicini.  
Coordinating activities and change: An event-driven architecture for situated MAS.  
*Engineering Applications of Artificial Intelligence*, January 2015.

# References VI



Andrea Omicini.

On the semantics of tuple-based coordination models.

In *1999 ACM Symposium on Applied Computing (SAC'99)*, pages 175–182, New York, NY, USA, 28 February – 2 March 1999. ACM.

Special Track on Coordination Models, Languages and Applications.



Andrea Omicini.

Towards a notion of agent coordination context.

In Dan C. Marinescu and Craig Lee, editors, *Process Coordination and Ubiquitous Computing*, chapter 12, pages 187–200. CRC Press, Boca Raton, FL, USA, October 2002.



Andrea Omicini, Alessandro Ricci, Mirko Viroli, Cristiano Castelfranchi, and Luca Tummolini.

Coordination artifacts: Environment-based coordination for intelligent agents.

In Nicholas R. Jennings, Carles Sierra, Liz Sonenberg, and Milind Tambe, editors, *3rd international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, volume 1, pages 286–293, New York, USA, 19–23 July 2004. ACM.



# References VII



Andrea Omicini, Alessandro Ricci, and Mirko Viroli.

Artifacts in the A&A meta-model for multi-agent systems.

*Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, December 2008.

Special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems.



Andrea Omicini, Alessandro Ricci, and Nicola Zaghini.

Distributed workflow upon linkable coordination artifacts.

In Paolo Ciancarini and Herbert Wiklicky, editors, *Coordination Models and Languages*, volume 4038 of *LNCS*, pages 228–246. Springer, June 2006.



George A. Papadopoulos and Farhad Arbab.

Coordination models and languages.

In Marvin V. Zelkowitz, editor, *The Engineering of Large Systems*, volume 46 of *Advances in Computers*, pages 329–400. Academic Press, 1998.



Peter R. Pietzuch and Jean M. Bacon.

Hermes: a distributed event-based middleware architecture.

In *22nd International Conference on Distributed Computing Systems Workshops (ICDCS 2002)*, pages 611–618, Vienna, Austria, 2–5 July 2002. IEEE Computer Society.

# References VIII



Lucy A. Suchman.

Situated actions.

In *Plans and Situated Actions: The Problem of Human-Machine Communication*, chapter 4, pages 49–67. Cambridge University Press, New York, NYU, USA, 1987.



Mirko Viroli and Andrea Omicini.

Coordination as a service.

*Fundamenta Informaticae*, 73(4):507–534, 2006.  
Special Issue: Best papers of FOCLASA 2002.



Mirko Viroli, Andrea Omicini, and Alessandro Ricci.

On the expressiveness of event-based coordination media.

In Hamid R. Arabnia, editor, *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02)*, volume III, pages 1414–1420, Las Vegas, NV, USA, July 2002. CSREA Press.



Mirko Viroli and Alessandro Ricci.

Tuple-based coordination models in event-based scenarios.

In *22nd International Conference on Distributed Computing Systems*, pages 595–601. IEEE CS, 2002.

Proceedings of the Workshops.

# References IX



Peter Wegner.

Coordination as constrained interaction.

In Paolo Ciancarini and Chris Hankin, editors, *Coordination Languages and Models. First International Conference, COORDINATION '96 Cesena, Italy, April 15–17, 1996. Proceedings*, Lecture Notes in Computer Science, pages 28–33, Cesena, Italy, April 1996. Springer Berlin Heidelberg.



Michael J. Wooldridge.

*An Introduction to Multi-Agent Systems.*

John Wiley & Sons Ltd., Chichester, UK, March 2002.



Danny Weyns, Andrea Omicini, and James J. Odell.

Environment as a first-class abstraction in multi-agent systems.

*Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, February 2007.

# URLs

## Slides

- On APICe

<http://apice.unibo.it/xwiki/bin/view/Talks/EbsmasCoordination2015>

- On SlideShare

[http://www.slideshare.net/andreaomicini/  
blending-eventbased-and-multiagent-systems-around-coordination](http://www.slideshare.net/andreaomicini/blending-eventbased-and-multiagent-systems-around-coordination)

## Paper

- On APICe

<http://apice.unibo.it/xwiki/bin/view/Publications/EbsmasCoordination2015>

- On SpringerLink

[http://link.springer.com/10.1007/978-3-319-19282-6\\_12](http://link.springer.com/10.1007/978-3-319-19282-6_12)

# Blending Event-Based and Multi-Agent Systems around Coordination Abstractions

Andrea Omicini, Giancarlo Fortino, Stefano Mariani

ALMA MATER STUDIORUM—Università di Bologna, Italy  
Università della Calabria, Italy

COORDINATION 2015  
Grenoble, France  
3 June 2015