Models of Autonomy and Coordination

Integrating Subjective & Objective Approaches in Agent Development Frameworks

Stefano Mariani, Andrea Omicini, Luca Sangiorgi

{s.mariani, andrea.omicini}@unibo.it, luca.sangiorgi6@studio.unibo.it

DISI
Alma Mater Studiorum—Università di Bologna

Talk @ *International Symposium on Intelligent Distributed Computing*
Autonomous University of Madrid - 3rd September 2014

1. Context, Motivation & Goals

2. Autonomy & Coordination: Models and Technologies
   - Autonomy & Coordination in Agent Development Frameworks
   - Autonomy & Coordination in Agent Infrastructures

3. Autonomy-Preserving Integration Approaches
   - Preserving Autonomy in TuCSoN4Jade
   - Showcasing TuCSoN4Jade: the "Book Trading" Example

4. Conclusion & Further Work

Outline

1. Context, Motivation & Goals

2. Autonomy & Coordination: Models and Technologies
   - Autonomy & Coordination in Agent Development Frameworks
   - Autonomy & Coordination in Agent Infrastructures

3. Autonomy-Preserving Integration Approaches
   - Preserving Autonomy in TuCSoN4JADE
   - Showcasing TuCSoN4JADE: the “Book Trading” Example

4. Conclusion & Further Work

Context

- In objective coordination [Omicini and Ossowski, 2003], coordination-related concerns are extracted from agents to be embodied within dedicated abstractions offering *coordination as a service* [Viroli and Omicini, 2006]
- In subjective coordination [Omicini and Ossowski, 2003] instead, coordination issues are directly tackled by individual agents themselves

Objective & Subjective Coordination

Objective and subjective coordination thus constitute two *complementary* approaches, *both* essential in MAS design and development [Ricci et al., 2003], hence requiring a suitable integration.

Motivation

- Successful integration depends on the *technology level*, that is, on the mechanisms provided by the agent frameworks to be integrated
- In particular, it depends on the model of autonomy promoted by the specific agent platform, and by its relationship with the model of coordination adopted by the specific (objective) coordination framework

Hindering Autonomy

Any integration effort *not* taking into account such two aspects is likely to hinder agent autonomy by (unintentionally) creating *artificial dependencies* between the subjective and the objective stances on coordination

Goals

1. Define what a *model of autonomy* and a *model of coordination*[1] is
2. Analyse agent development frameworks and coordination infrastructures to understand the models of autonomy and coordination they adopt—either *implicitly* or *explicitly*
3. Provide an example of autonomy-preserving integration of objective and subjective coordination by discussing TuCSoN4JADE (`http://bitbucket.org/smariani/tucson/wiki/Home`)
4. Demonstrate practical consequences of such integration by comparison with a *non autonomy-preserving* integration attempt [Omicini et al., 2004]

[1] Don't be confused with the notion of *coordination model* [Ciancarini, 1996]

Outline

1. Context, Motivation & Goals

2. Autonomy & Coordination: Models and Technologies
    - Autonomy & Coordination in Agent Development Frameworks
    - Autonomy & Coordination in Agent Infrastructures

3. Autonomy-Preserving Integration Approaches
    - Preserving Autonomy in TuCSoN4Jade
    - Showcasing TuCSoN4Jade: the “Book Trading” Example

4. Conclusion & Further Work

Models of Autonomy and Coordination

Model of Autonomy

A model defining *(i)* how agents behave as *individual* entities, *(ii)* how they relate to each other as *social* entities, as well as *(iii)* how the two things *coexist*.

Model of Coordination

A model defining the semantics of the admissible *interactions* between agents in a MAS, in particular, w.r.t. their effects on the agent autonomy (e.g., *control flow*).

Outline

1. Context, Motivation & Goals

2. Autonomy & Coordination: Models and Technologies
   - Autonomy & Coordination in Agent Development Frameworks
   - Autonomy & Coordination in Agent Infrastructures

3. Autonomy-Preserving Integration Approaches
   - Preserving Autonomy in TuCSoN4Jade
   - Showcasing TuCSoN4Jade: the "Book Trading" Example

4. Conclusion & Further Work

JADE [Bellifemine et al., 1999] I

- Autonomy of agents is supported by the behaviours mechanism, whereas their mutual interaction by the Agent Communication Channel (ACC)

Behaviours

- Logically, a behaviour is "an activity to perform with the goal of accomplishing a task"
- Technically, behaviours are Java objects executed pseudo-concurrently within a *single Java thread* by a non-preemptive round-robin scheduler

- JADE programmers implement within the `action()` method of each behaviour one of the agent activities composing its course of actions
- Such method is executed from the beginning every time: there is *no way to "stop-then-resume"* a behaviour

JADE [Bellifemine et al., 1999] II

The Agent Communication Channel (ACC)

The ACC is the run-time service in charge of asynchronous message passing among agents: each agent has its own mailbox, and *is notified* upon reception of any message.

- JADE programmers can retrieve messages either *asynchronously* (`receive()`) or *synchronously* (`blockingReceive()`)
- `blockingReceive()` suspends the agent, not only the calling behaviour
- Such semantics impacts the aforementioned model of autonomy

Hindering Autonomy

Synchronous communication hinders autonomy of the caller agent, since *all its other behaviours* – not just the caller one – are suspended by the communication semantics.

JADE [Bellifemine et al., 1999] III

- To preserve agents autonomy, the JADE Programmers Guide [Bellifemine et al., 2002] suggests calling `receive()` instead, then `block()` if no message is found, so as to let JADE runtime *suspend only the calling behaviour*

JADE Model of Autonomy

Summing up:

- behaviours for individual tasks
- asynchronous messages for subjective coordination
- the "`block()`-then-resume" pattern to reconcile individual and social attitudes

Jason [Bordini et al., 2007] I

- Autonomy of agents is supported by the plan/intention execution machinery and the message passing service

Plans/Intentions

- Logically, a plan is "an activity to perform with the goal of accomplishing a task"
- Technically, plans are BDI structures which are firstly scheduled, then instantiated as *intentions*, finally executed pseudo-concurrently *one action each*, according to a round-robin scheduler

- Intentions may be *suspended* by the Jason reasoner for a number of reasons, e.g. because the agent needs to wait for a message

Jason [Bordini et al., 2007] II

Message Passing

Jason agents can *exchange beliefs/plans/goals* in the form of messages.

- Intentions are *automatically suspended* whenever they perform a *communication action* which cannot complete, to be later resumed as soon as the action obtains its completion feedback [Bordini et al., 2007]
- This preserves Jason agent autonomy similarly to the "`block()`-then-resume" pattern in JADE

Jason Model of Autonomy

Summing up:

- plans/intentions for individual tasks
- asynchronous message passing for subjective coordination
- intention suspension to reconcile individual and social attitudes

Outline

1. Context, Motivation & Goals

2. Autonomy & Coordination: Models and Technologies
   - Autonomy & Coordination in Agent Development Frameworks
   - Autonomy & Coordination in Agent Infrastructures

3. Autonomy-Preserving Integration Approaches
   - Preserving Autonomy in TuCSoN4Jade
   - Showcasing TuCSoN4Jade: the “Book Trading” Example

4. Conclusion & Further Work

TuCSoN I

- TuCSoN [Omicini and Zambonelli, 1999] is a Java-based, tuple-based coordination model and infrastructure for open, distributed MAS
- Its model of coordination is supported by Agent Coordination Contexts (ACC) [Omicini, 2002]

Agent Coordination Contexts

ACCs are “mediators” *enabling and constraining* agents interaction, mapping coordination operations into events and asynchronously dispatching them to the coordination medium.

- ACCs are fundamental to *guarantee and preserve* agent autonomy by enabling separation of the suspensive semantics of a coordination operation from its invocation semantics

TuCSoN II

Operation Execution

invocation — the *request* to carry out a given coordination operation is sent to TuCSoN

completion — the *response* to the coordination operation invoked is sent back to the requesting agent

- Thus, any coordination operation in TuCSoN is *asynchronous by default*
- Nevertheless, agents freely choose invocation semantics to be either *synchronous* or *asynchronous*—*regardless* of the suspensive semantics of the operation (e.g., `in` vs. `inp`)

TuCSoN III

TuCSoN Model of Coordination

By decoupling invocation semantics from the operation semantics, synchronous calls are always consequence of the *agent own deliberation* process.

CArtAgO I

- CArtAgO [Ricci et al., 2007] is a Java-based framework and infrastructure based on the A&A (agents & artefacts) meta-model [Omicini et al., 2008], exploiting *artefacts* as the tools that agents use to achieve their own goals
- Its model of coordination is based on the agent body abstraction

Agent Bodies

By exposing an *effectors* API and a *perception* API, CArtAgO agent bodies are the architectural components *enabling* (and decoupling) agent interactions with artefacts.

- When acting on effectors, only the current agent activity is *suspended* until action *completion* is reported—not the agent as a whole

CArtAgO II

CArtAgO Model of Coordination

Mediation by agent bodies is the mechanism preserving agent autonomy in CArtAgO by *uncoupling* action suspension from caller agent suspension.

Outline

1. Context, Motivation & Goals

2. Autonomy & Coordination: Models and Technologies
   - Autonomy & Coordination in Agent Development Frameworks
   - Autonomy & Coordination in Agent Infrastructures

3. Autonomy-Preserving Integration Approaches
   - Preserving Autonomy in TuCSoN4JADE
   - Showcasing TuCSoN4JADE: the "Book Trading" Example

4. Conclusion & Further Work

An Example of Successful Integration

- In [Ricci et al., 2008a], CArtAgO is integrated with Jason [Bordini et al., 2007]
- Although the goal was to promote artefact-based interaction of heterogeneous agents, authors *de facto* integrate message-based (subjective) coordination with artefact-based (objective) coordination, by allowing Jason agents to exploit CArtAgO for building & using coordination artefacts

Autonomy Unhindered

Whenever an agent requests execution of an operation on an artefact, *the caller intention* (only) is automatically suspended until the "effector feedback" is received. Thus, nothing can hinder agents autonomy if they *simultaneously* operate on artefacts while exchanging messages.

An Example of Unsuccessful Integration

- In [Omicini et al., 2004], integration between JADE and TuCSoN technologies is achieved, allowing JADE agents to exploit TuCSoN coordination services as part of the JADE platform

Autonomy Hindered

However, JADE model of autonomy and TuCSoN model of coordination *were not considered*: whenever a coordination operation gets stuck, the caller behaviour unavoidably gets stuck, too, because of its single thread of control being waiting for operation completion.

- The agent choice to rely on objective coordination affects its ongoing subjective coordination activities
- This is a clear example of an *artificial dependency* (unintentionally) created by a “non autonomy-preserving” approach

Outline

1. Context, Motivation & Goals
2. Autonomy & Coordination: Models and Technologies
   - Autonomy & Coordination in Agent Development Frameworks
   - Autonomy & Coordination in Agent Infrastructures
3. Autonomy-Preserving Integration Approaches
   - Preserving Autonomy in TuCSoN4JADE
   - Showcasing TuCSoN4JADE: the "Book Trading" Example
4. Conclusion & Further Work

TuCSoN4JADE I

- TuCSoN4JADE preserves autonomy of agents by providing them with two invocation semantics regarding coordination services:

  `synchronousInvocation()` — lets agents invoke TuCSoN coordination operations synchronously w.r.t. the caller behaviour. This means the caller behaviour *only* is (possibly) suspended and automatically resumed.

  `asynchronousInvocation()` — lets clients *asynchronously* invoke TuCSoN coordination operations. Regardless of whether the coordination operation suspends, the agent does not, thus the caller behaviour continues.

- The method for synchronous invocation of coordination services is the one we are interested in

TuCSoN4JADE II

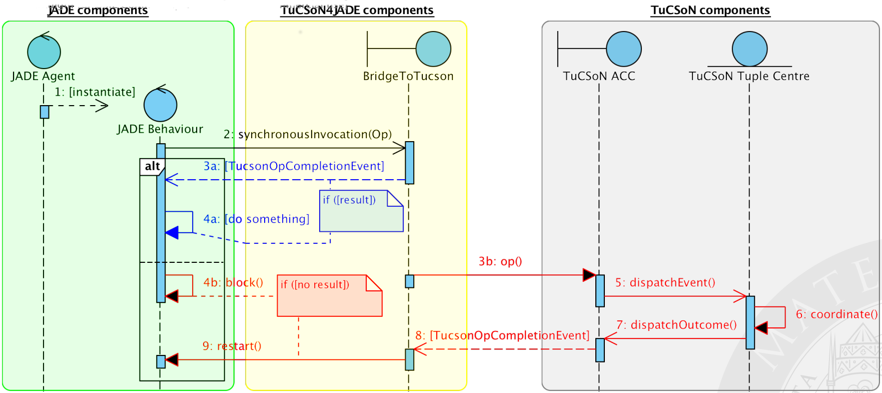

Figure : The "alt"-labelled frame is the equivalent of JADE `blockingReceive()` programming pattern in TuCSoN4JADE.

TuCSoN4JADE III

- We know when JADE behaviour is re-scheduled, its `action()` method re-starts *from the beginning*, thus, method `synchronousInvocation()` is re-invoked

Autonomy Unhindered

The whole TuCSoN4JADE machinery works because such method internally (thus *transparently*) checks if operation completion is already available: only if it is not, the behaviour (only) gets suspended, thus the whole path 3.b-9 executed.

- This way, the agents choice to rely on objective coordination *no longer affects* their ongoing subjective coordination activities
- This has been possible by accounting for JADE model of autonomy and TuCSoN model of coordination while planning integration

Outline

1. Context, Motivation & Goals
2. Autonomy & Coordination: Models and Technologies
   - Autonomy & Coordination in Agent Development Frameworks
   - Autonomy & Coordination in Agent Infrastructures
3. Autonomy-Preserving Integration Approaches
   - Preserving Autonomy in TuCSoN4JADE
   - Showcasing TuCSoN4JADE: the "Book Trading" Example
4. Conclusion & Further Work

Scenario I

- *n* seller agents advertise their catalogue of books
- *m* buyer agents browse such catalogues looking for books
- The whole interactions chain has the form of the well-known *ContractNet protocol*:
  1. buyers start a call-for-proposals
  2. sellers reply with actual proposals
  3. buyers choose which one to accept
  4. the purchase is carried out

Concurrency Property

Sellers should stay reactive to call-for-proposals even in the middle of a purchase transaction—otherwise they could lose potential revenues. We call concurrency property such a requirement.

Scenario II

- We re-think the ContractNet protocol by integrating objective and subjective coordination: tuple-based call-for-proposals (thus, objective coordination) with message-based purchase (hence, subjective coordination)
- The call-for-proposals should reach all the sellers, thus it is more efficient to put a single "call-for-proposals tuple" in a shared "contract-net space", rather than messaging each seller individually
- The purchase is typically a 1-to-1 interaction, hence messaging can efficiently do the job
- We compare the integration of TuCSoN and JADE proposed in [Omicini et al., 2004] with ours using TuCSoN4JADE (the code is available as part of the TuCSoN4JADE distribution, downloadable from `http://bitbucket.org/smariani/tucson/downloads`)
- As expected, in the former case the concurrency property is lost, whereas in the latter it is preserved

[Omicini et al., 2004] Approach

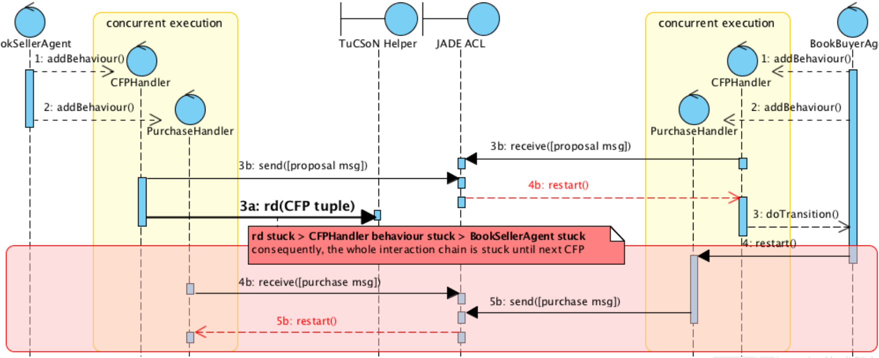

Figure : The rd is stuck on a network-level call, thus the caller behaviour is stuck too, hindering the caller agent from scheduling other behaviours in the meanwhile—in particular, the "purchase" interaction chain (4b-5b) cannot carry on until a new call-for-proposals is issued.

TuCSoN4JADE Approach

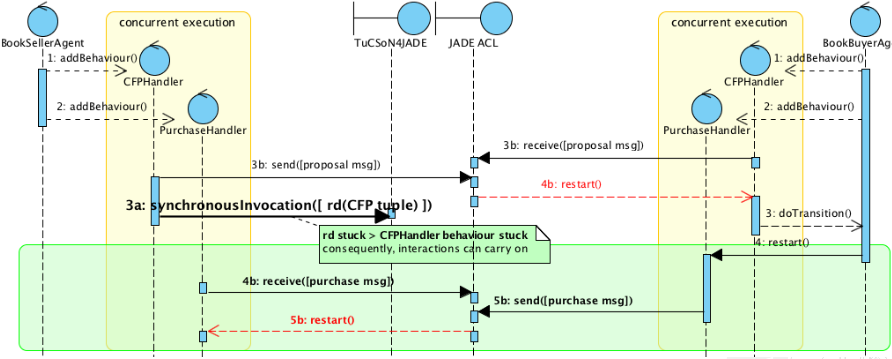

Figure : The rd suspensive semantics is confined to the caller behaviour, thus only the caller behaviour is suspended, whereas other activities can carry on concurrently—e.g., the purchase transaction already in place (4b-5b).

Outline

1. Context, Motivation & Goals

2. Autonomy & Coordination: Models and Technologies
   - Autonomy & Coordination in Agent Development Frameworks
   - Autonomy & Coordination in Agent Infrastructures

3. Autonomy-Preserving Integration Approaches
   - Preserving Autonomy in TuCSoN4JADE
   - Showcasing TuCSoN4JADE: the "Book Trading" Example

4. Conclusion & Further Work

Conclusion

- The main point was not to show how JADE and TuCSoN are now better integrated w.r.t. [Omicini et al., 2004]
- Instead, we aim at stressing how technology-level details may have deep consequences on the *higher levels of abstraction*, whenever the models (possibly *implicitly*) brought about by technologies are not properly accounted for and understood

Impact on Autonomy

In particular, we demonstrate how the models of autonomy and coordination promoted by agent development frameworks *may hamper* an essential feature of agents: autonomy.

Further Work

- We believe the issue of autonomy-preserving approaches in integrating subjective and objective coordination is quite a general one
- Thus, further work will be devoted to analyse and integrate other frameworks—e.g. building TuCSoN4Jason, TuCSoN4simpA [Ricci et al., 2008b], etc.

Thanks

Thank you for your attention

References I

Bellifemine, F., Caire, G., Trucco, T., and Rimassa, G. (2002).
Jade programmer's guide.
Jade version, 3.

Bellifemine, F. L., Poggi, A., and Rimassa, G. (1999).
JADE–a FIPA-compliant agent framework.
In *4th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-99)*, pages 97–108, London, UK. The Practical Application Company Ltd.

Bordini, R. H., Hübner, J. F., and Wooldridge, M. J. (2007).
Programming Multi-Agent Systems in AgentSpeak using Jason.
John Wiley & Sons.

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

Omicini, A. (2002).
Towards a notion of agent coordination context.
In Marinescu, D. C. and Lee, C., editors, *Process Coordination and Ubiquitous Computing*, chapter 12, pages 187–200. CRC Press, Boca Raton, FL, USA.

References II

Omicini, A. and Ossowski, S. (2003).
Objective versus subjective coordination in the engineering of agent systems.
In Klusch, M., Bergamaschi, S., Edwards, P., and Petta, P., editors, *Intelligent Information Agents: An AgentLink Perspective*, volume 2586 of *LNCS*, pages 179–202. Springer.

Omicini, A., Ricci, A., and Viroli, M. (2008).
Artifacts in the A&A meta-model for multi-agent systems.
Autonomous Agents and Multi-Agent Systems, 17(3):432–456.

Omicini, A., Ricci, A., Viroli, M., Cioffi, M., and Rimassa, G. (2004).
Multi-agent infrastructures for objective and subjective coordination.
Applied Artificial Intelligence, 18(9-10):815–831.

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

Ricci, A., Omicini, A., and Denti, E. (2003).
Activity Theory as a framework for MAS coordination.
In Petta, P., Tolksdorf, R., and Zambonelli, F., editors, *Engineering Societies in the Agents World III*, volume 2577 of *LNCS*, pages 96–110. Springer.

References III

Ricci, A., Piunti, M., Acay, L. D., Bordini, R. H., Hübner, J., and Dastani, M. (2008a).
Integrating artifact-based environments with heterogeneous agent-programming platforms.
In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-08)*, pages 225–232, Estoril, Portugal. IFAAMAS.

Ricci, A., Viroli, M., and Omicini, A. (2007).
CArtAgO: A framework for prototyping artifact-based environments in MAS.
In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNAI*, pages 67–86. Springer.

Ricci, A., Viroli, M., and Piancastelli, G. (2008b).
simpA: A simple agent-oriented Java extension for developing concurrent applications.
In *Languages, Methodologies and Development Tools for Multi-Agent Systems*, volume 5118 of *LNCS*, pages 261–278. Springer.

Viroli, M. and Omicini, A. (2006).
Coordination as a service.
Fundamenta Informaticae, 73(4):507–534.

Models of Autonomy and Coordination

Integrating Subjective & Objective Approaches in Agent Development Frameworks

Stefano Mariani, Andrea Omicini, Luca Sangiorgi

{s.mariani, andrea.omicini}@unibo.it, luca.sangiorgi6@studio.unibo.it

DISI
Alma Mater Studiorum—Università di Bologna

Talk @ *International Symposium on Intelligent Distributed Computing*
Autonomous University of Madrid - 3rd September 2014