

# Operating instructions for intelligent agent coordination

a paper by Mirko Viroli, Alessandro Ricci and Andrea Omicini

Università de Bologna

Nikola Ciprich - VŠB Technical University of Ostrava

December 8, 2007

- 1 Introduction
  - The goal
- 2 Environment based coordination
  - Coordination artifacts
  - An abstract model
- 3 Application examples
  - A request/response scenario
  - A Contract-Net scenario
  - TuCSoN
- 4 Conclusion

# The goal

- Explore possibilities of *environment based coordination*
- Introduce *coordination artifacts*
- Propose *operating instructions* for using them

# Environment based coordination

- Agents do not always communicate directly
- Agents communication is not based only on their implemented behaviour
- Usage of *coordination artifacts*

# Coordination artifact

- Special mediating agent providing services for other agents coordination

# Coordination artifact

- Special mediating agent providing services for other agents coordination
- Does not need to be an autonomous agent

## Coordination artifact

- Special mediating agent providing services for other agents coordination
- Does not need to be an autonomous agent
- Exports specific *usage interface* - has a set of an “operating instructions”

# Coordination artifact

- Special mediating agent providing services for other agents coordination
- Does not need to be an autonomous agent
- Exports specific *usage interface* - has a set of an “operating instructions”
- Is never proactive - actions are executed by agents



# Coordination artifact

- Special mediating agent providing services for other agents coordination
- Does not need to be an autonomous agent
- Exports specific *usage interface* - has a set of an “operating instructions”
- Is never proactive - actions are executed by agents
- Programmable - coordination can be fully dynamic

# An abstract model

- The basic abstract model of coordination artifact:
  - a *usage interface*
  - a set of *operating instructions*
  - a *coordination behaviour specification*

## Application examples

- A request/response scenario
- A contract-Net scenario

# A request/response scenario

- Resembles a *query* FIPA performative

## A request/response scenario

- Resembles a *query* FIPA performative
- Coordination artifact supports two agent roles:
  - a *client* agent interested in some information
  - a *server* agent able to provide the information

## A request/response scenario

- Resembles a *query* FIPA performative
- Coordination artifact supports two agent roles:
  - a *client* agent interested in some information
  - a *server* agent able to provide the information
- and provides three actions:
  - *ask* - used by client to query the artifact for information
  - *get* - used by a server to ask for pending queries
  - *tell* - used by a server to provide information

## A Contract-Net scenario

- Coordination artifact supports two agent roles:
  - an *initiator* agent - seeks the service
  - a set of *participants* - can provide the service

## A Contract-Net scenario

- Coordination artifact supports two agent roles:
  - an *initiator* agent - seeks the service
  - a set of *participants* - can provide the service
- Actions for initiator agent:
  - *CFP* - used to initiates the call for proposals
  - *getProp* - used to get submitted proposal
  - *acceptProp*, *refuseProp* - used to accept/reject the proposal



## A Contract-Net scenario

- Coordination artifact supports two agent roles:
  - an *initiator* agent - seeks the service
  - a set of *participants* - can provide the service
- Actions for initiator agent:
  - *CFP* - used to initiates the call for proposals
  - *getProp* - used to get submitted proposal
  - *acceptProp*, *refuseProp* - used to accept/reject the proposal
- Actions for participant agents:
  - *getCFP* - used to get pending CFPs
  - *refuse* - used to refuse answering CFP
  - *propose* - used to submit the proposal

## Coordination artifacts in practice - *TuCSoN*

- *TuCSoN* agents interact using *tuples*
- Coordination artifacts are called *tuple centers*
- Tuple centers can be accessed using simple communication operations (*out*, *rd*, *in*, ...)
- <http://tucson.sourceforge.net>

# Conclusion

- Semantic framework for interaction of agents with coordination artifacts has been presented

# Conclusion

- Semantic framework for interaction of agents with coordination artifacts has been presented
- Future work:
  - Extension of operating instructions with notion of timeout violations and guarantees
  - Integration with existing MAS frameworks

# Discussion

- Questions?

# Discussion

- Questions?
- Thanks for Your attention