The Distributed Autonomy

Software Abstractions and Technologies for Autonomous Systems

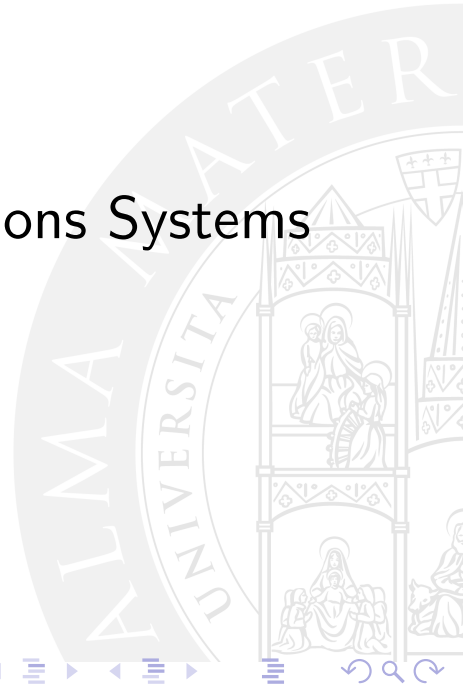
Andrea Omicini

Alma Mater Studiorum—Università di Bologna

CCW Meeting of Experts on Lethal Autonomous Weapons Systems

United Nations, Genève, Switzerland

13 April 2015



Autonomous Systems as Software Systems

- Artificial systems in general feature nowadays an ever-growing relevance of ICT components and models
- When *autonomy* is concerned, issues like deliberation, planning, knowledge representation, and the like, emphasise the role of computational/software component/subsystems
- In perspective, talking about forthcoming *autonomous systems* mostly means talking about *software systems* / components

Complex Software Systems as Multi-Agent Systems [ZO04]

- Nowadays, most of the complex computational systems of interest can be thought, modelled, and built as multi-agent systems (MAS)
- MAS are not necessarily autonomous; however
  - they are built out of many autonomous components, called agents
  - they are the reference computational paradigm for building autonomous (software) systems

Socio-Technical Systems as MAS

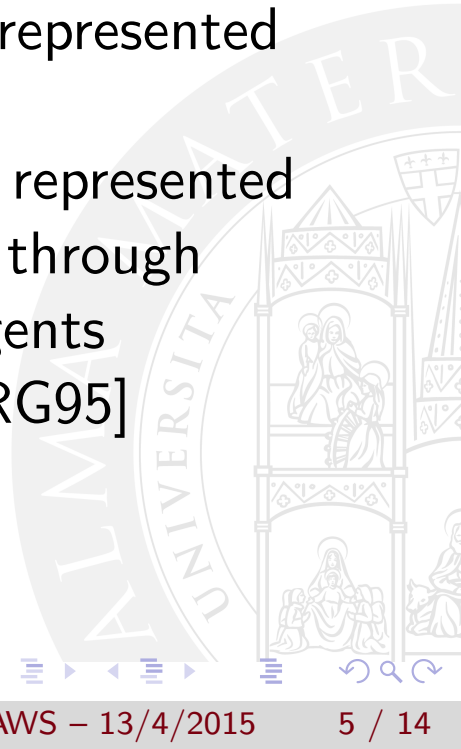
- Most of the relevant systems nowadays are socio-technical systems—that is, systems where components are human and software agents altogether
  - as AWS typically are
- When modelling and engineering socio-technical systems, the agent abstraction typically accounts for *both* human and software agents

Agents & Goals

- Agents are computational entities whose defining feature is *autonomy* [Woo02]
- Agents are *goal-driven*, since goals determine and explain the agent's course of actions [CC95]

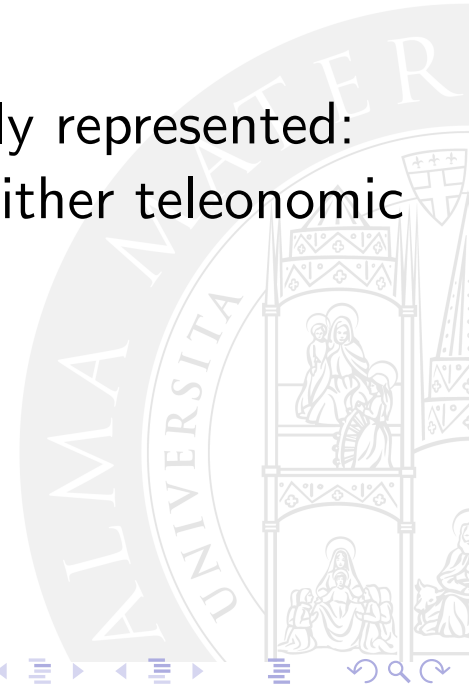
  teleonomic (goal-oriented) agents feature implicitly represented goals (*weak agency*)

  teleologic (goal-governed) agents feature explicitly represented goals (*strong agency*), typically handled through *mentalistic abstractions* by intelligent agents [WJ95]—e.g. BDI agent architectures [RG95]



Agent Societies & Coordination

- Agent societies rule collective MAS behaviours towards the overall system goals, by governing mutual agent *dependencies* [MC94].
- Agent societies are built around coordination media [GC92], encapsulating social (coordination) laws
- There, social goals may be either implicitly or explicitly represented: so, in turn, societies (and MAS in general) could be either teleonomic or teleologic, respectively, as wholes



Distributed Autonomy

- When a complex socio-technical system (such as an AWS) is built as a MAS
  - a number of autonomous components (either humans or software agents) and structures (societies) are in place
  - each one capable to pursue its own goals either teleonomically or teleologically
- So, autonomy could be conceived as a *distributed property* of socio-technical systems
  - distributed autonomy

Critical Issues I

Who is in charge, really?

- Distributed autonomy means that *decisions* are actually *distributed*
  - possibly with components featuring different *sorts of autonomy* in the same system
  - possibly distributed among both human and software agents in an articulated way
  - possibly in a dynamic way, at run time
- Autonomy, deliberation, decision: it is no longer like pulling a trigger
- *It is much more complex than that*
- Autonomy is distributed
  - and so are responsibility and liability

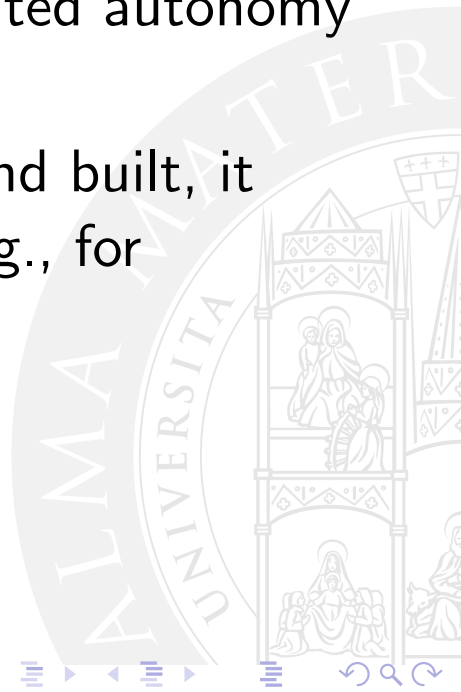
Critical Issues II

Further sources of complexity

- Teleonomic / teleologic agents / societies typically coexist in the same MAS
- In critical socio-technical systems, any sort of deliberation (human included) typically depend on huge amounts of data and information elaborated by (possibly autonomous) software components
- Agents may depend on each others, *interfere*, exchange goals
- Any agent may belong to more than one MAS, and make different systems interfere with each other
- *Self-organising MAS* make it possible to build autonomous systems – including LAWS – which are not just teleonomic, but also has no single place for system goals—goals are nowhere *visible* when observing the system

Norms for LAWS?

- Understanding who/what is actually taking a decision – and, based on what – is no longer a trivial issue when distributed autonomy is in place
- Without a well-founded engineering discipline, distributed autonomy may lead to *uncertain responsibility* / liability
- Without norms on how LAWS are actually designed and built, it unclear whether LAWS could be actually regulated, e.g., for compliancy with IHL principles



Bibliography I

Rosaria Conte and Cristiano Castelfranchi, editors.
Cognitive and Social Action.
Routledge, 1995.

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

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

Bibliography II

Anand S. Rao and Michael P. Georgeff.
BDI agents: From theory to practice.
In Victor R. Lesser and Les Gasser, editors, *1st International Conference on Multi Agent Systems (ICMAS 1995)*, pages 312–319, San Francisco, CA, USA, 12-14 June 1995. The MIT Press.

Michael J. Wooldridge and Nicholas R. Jennings.
Intelligent agents: Theory and practice.
Knowledge Engineering Review, 10(2):115–152, June 1995.

Michael J. Wooldridge.
An Introduction to MultiAgent Systems.
John Wiley & Sons Ltd., Chichester, UK, March 2002.

Bibliography III

Franco Zambonelli and Andrea Omicini.
Challenges and research directions in agent-oriented software engineering.
Autonomous Agents and Multi-Agent Systems, 9(3):253–283, November 2004.
Special Issue: Challenges for Agent-Based Computing.

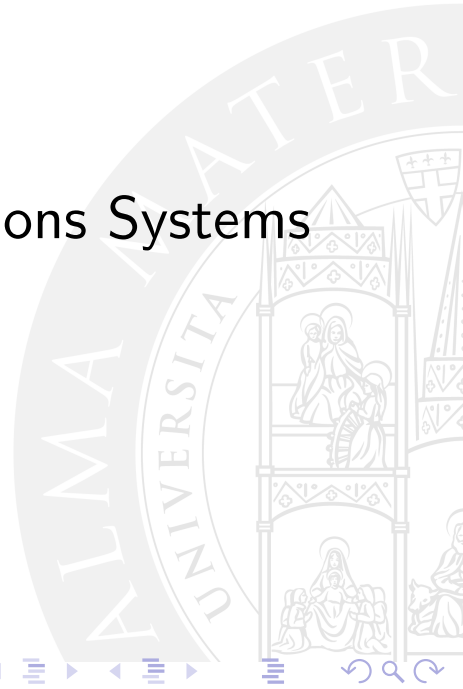
The Distributed Autonomy

Software Abstractions and Technologies for Autonomous Systems

Andrea Omicini

Alma Mater Studiorum—Università di Bologna

CCW Meeting of Experts on Lethal Autonomous Weapons Systems

United Nations, Genève, Switzerland

13 April 2015