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, Géneve, Switzerland
13 April 2015
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.
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.
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 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 (\textit{weak agency})
- **teleologic** (goal-governed) agents feature explicitly represented goals (\textit{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
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
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.


BDI agents: From theory to practice.  

Michael J. Wooldridge and Nicholas R. Jennings.  
Intelligent agents: Theory and practice.  

Michael J. Wooldridge.  
*An Introduction to MultiAgent Systems*.  
John Wiley & Sons Ltd., Chichester, UK, March 2002.
Franco Zambonelli and Andrea Omicini. Challenges and research directions in agent-oriented software engineering.


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, Géneve, Switzerland
13 April 2015