

Specifying Open Agent Systems: A Survey

Alexander Artikis^{1 2} and Jeremy Pitt²

¹Software & Knowledge Engineering Laboratory,
NCSR “Demokritos”

²Electrical & Electronic Engineering Department,
Imperial College London

September 24, 2008

Open Agent Systems

Multi-agent systems that exhibit the following characteristics:

- ▶ The **internal** architectures of the members are **not publicly known**.
- ▶ The members of the system are **self-interested**; they do not necessarily share a notion of a global utility.
- ▶ The members' interactions are always subject to **unanticipated outcomes**.

Actuality (what is the case) does not necessarily coincide with ideality (what ought to be the case).

Entering and leaving an OAS via a role-management procedure.

Examples: E-markets, DRM, VO, p2p, AHN, ...

A Survey of Open Agent System Specifications

Approaches:

- ▶ Artificial Social Systems.
- ▶ Enterprise Modelling.
- ▶ Commitment Protocols.
- ▶ Electronic Institutions.

Evaluation criteria:

- ▶ Normative relations.
- ▶ Formal semantics.
- ▶ Declarative semantics.
- ▶ Verifiable semantics.
- ▶ Computational support.

Artificial Social Systems

Social law (α, φ) *prohibits* agent i to perform action α when the system is in state w provided that $w_i \models \varphi$.

Every physically possible action that is not prohibited by a social law is considered *allowed* by that law.

- ▶ Every allowed action is physically possible.
- ▶ In every state, there is at least one action an agent is allowed to perform.

A social system is a system in which the initial states are 'socially acceptable' and every agent obeys the social laws.

Artificial Social Systems: Commentary

- ▶ Social laws only refer to permissions/prohibitions.
- ▶ There are no mechanisms for dealing with law violation.
- ▶ Formal, declarative semantics. Verification of compliance requires that φ in social laws (α, φ) does not refer to the internals of the agents.
- ▶ There is no computational support.

Enterprise Modelling

An organisation is a set of agents playing roles in which they are acting to achieve specific goals according to the 'rules of the game'.

These express the permissions, obligations, commitments, authorities, powers of the organisation agents.

Depending on the type of authority Ag_1 has over Ag_2 , due to the roles Ag_1 and Ag_2 occupy:

- ▶ Ag_1 may assign a set of goals to Ag_2 .
- ▶ Ag_1 may assign a set of new roles to Ag_2 .
- ▶ Ag_1 may empower Ag_2 to perform a set of actions.

The rules of the game are expressed in the Situation Calculus. A logic programming implementation of the Situation Calculus supports planning and narrative assimilation.

Enterprise Modelling: Commentary

- ▶ The representation of 'empowerment' is not clear.
- ▶ There are no mechanisms for dealing with rule violation.
- ▶ Formal, declarative, verifiable semantics.
- ▶ No discussion on the complexity of the supported computational tasks.

Commitment Protocols

Commitment as a directed, contextual obligation.

Commitment protocols: creating, discharging, cancelling, releasing, delegating, assigning commitments.

Various formalisations of commitment protocols, eg:

- ▶ Event Calculus. Computational support by an abductive Event Calculus planner.
- ▶ Action language $C+$. Computational support by the Causal Calculator (SAT-based implementation).
- ▶ Web Ontology Language for Protocols (OWL-P) and Semantic Web Rule Language (SWRL) aiming at protocol composition.

Commitment Protocols: Commentary

- ▶ Interaction protocols are specified only in terms of commitments. Other normative relations are at least as important.
- ▶ There are no mechanisms for dealing with commitment violation.
- ▶ Formal, declarative, verifiable semantics.
- ▶ Computational support:
 - ▶ It is assumed that the domain of each variable is finite and known from the outset.
 - ▶ The employed implementations can become inefficient when considering large systems.

Electronic Institutions

The core notions of an e-institution are: role, dialogic framework, scene, performative structure and normative rule.

Normative rules:

- ▶ 'Strict' obligations. No deontic operator is used in the representation.
- ▶ 'Less strict' rules expressing permissions and obligations with a temporal and/or conditional constraint.

Mechanisms for rule violation: regimentation devices, sanction.

Computational support:

- ▶ A development environment for specifying, and proving properties of, e-institutions.
- ▶ A rule-based system for the provision of run-time services.

Electronic Institutions: Commentary

- ▶ Interaction protocols are specified only in terms of permissions and obligations. Other normative relations are at least as important.
- ▶ Formal, declarative, verifiable semantics.
- ▶ The representation of the agents' actions and their effects is not as expressive as those of enterprise modelling and commitment protocols.
- ▶ Computational support:
 - ▶ Equivalence between (fragments of) the formalism used for proving properties with that used for providing run-time services has not been established.
 - ▶ The module for proving properties operates under the assumption that the domain of each variable is finite and known from the outset, while the complexity is exponential to the number of variables and their possible values.

Summary

“ ‘[E]mpowering’ is not an *exclusively* legal phenomenon, but is a standard feature of any norm-governed organisation where selected agents are assigned specific roles (in which they are empowered to conduct the business of that organisation).” [Jones & Sergot]

Mechanisms for dealing with rule violation.

Formal, declarative, verifiable semantics.

Computational support for design-time and run-time activities.

Specifying Open Agent Systems: A Survey

Thank you!

Questions?