AGENTS AND AGREEMENT TECHNOLOGIES: THE NEXT GENERATION OF DISTRIBUTED SYSTEMS

Vicent J. Botti Navarro
Grupo de Tecnología Informática- Inteligencia Artificial
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
From Prehistory to the Present

- With the hardware evolution software has progressed through:
From Prehistory to the Present

- With the hardware evolution software has progressed through:
  - machine code;
  - assembly language;
From Prehistory to the Present

- With the hardware evolution software has progressed through:
  - machine code;
  - assembly language;
  - machine-independent programming languages;
  - sub-routines;
  - procedures & functions;
    - Structured Programming
  - abstract data types;
    - Software Engineering
From Prehistory to the Present

- With the hardware evolution software has progressed through:
  - machine code;
  - assembly language;
  - machine-independent programming languages;
  - sub-routines;
  - procedures & functions;
    - Structured Programming
  - abstract data types;
    - Software Engineering
  - objects;
    - Internet
From Prehistory to the Present

With the hardware evolution software has progressed through:

- machine code;
- assembly language;
- machine-independent programming languages;
- sub-routines;
- procedures & functions;
  - Structured Programming
- abstract data types;
  - Software Engineering
- objects;
  - Internet

To

- Agents / Multi-agent Systems / Services
  - Agent Based Software Engineering
  - Service Oriented Software Engineering
Where are We?

- The recent years of technological evolution in the areas of Computer Technology and Communications (Internet, WWW, e-commerce, wireless connectivity, etc.) has given rise to new computation paradigms such as:
  - Peer-to-peer technologies,
  - Grid Computing,
  - Autonomic Computing
  - Computation as interaction.

- With these new paradigms it is becoming increasingly natural to view large systems in terms of the services they offer, and consequently in terms of the entities or agents providing or consuming services.
Computing as interaction

- In this new paradigm, computation is something that occurs by means of and through communication among computational entities.

- Computation becomes an inherently social activity, instead of solitary, leading to new forms of conceiving, designing, developing and handling computational systems.

- An example of the influence of this point of view is the emerging model of the software as a service, as in the service-oriented architectures.

- In this model, the applications are no longer monolithic single-user applications, or distributed applications managed by only one organization, but rather societies of computational entities (components) that can be conceived as service providers, they may enter or abandon different societies in different moments and for different reasons; and they may form coalitions or virtual organizations between themselves to attain their current goals.

- Then distributed systems are increasingly viewed as collections of service provider and service consumer components interlinked by dynamically defined workflows. Services must thus be realised by concrete entities or agents that send and receive messages, but the services themselves are the resources characterised by the functionality provided.

- Interconnection and Distribution, coupled with the need for systems to represent our best interests, implies systems that can cooperate and reach agreements (or even compete) with other systems that have different interests (much as we do with other people)
Virtual Organizations: A Problem Scenario
Virtual Organization

coordination

heterogeneity
Organizational Concepts

ROLES

NORMS

SERVICES
These emerging domains and environments are open and dynamic so that new agents may join and existing ones leave.

Agents act on behalf of service owners, managing access to services, and ensuring that contracts are fulfilled.

Agents act on behalf of service consumers, locating services, agreeing contracts, and receiving and presenting results.

Agents are required to engage in interactions, negotiate with one another, make agreements, and make proactive run-time decisions, individually and collectively, while responding to changing circumstances.

Agents need to collaborate and to form coalitions of agents with different capabilities in support of new virtual organisations.
What is needed for the ‘Computation as Interaction Paradigm’?

**Agreement Technologies**

We need develop models, frameworks, methods and algorithms for constructing large-scale open distributed computer systems.

The next generation computing systems needs autonomy, interaction and mobility as key issues.

We have to develop technologies to cope with the (high) dynamicity of the system topology and with semantic mismatches in the interaction, both natural consequences of the distributed and autonomous nature of the components.

We have focus on security issues by developing a new concept of operating system that incorporates low-level security mechanisms and trust measures that complement the classical cryptographic methods.

Trust measures are essential in open environments where interactions have to be made under uncertainty on the environment state.
What is needed for the ‘Computation as Interaction Paradigm’?

**Agreement Technologies (2)**

We need techniques that enable software components to reach agreements on the mutual performance of services.

Negotiation, argumentation, decision making, knowledge modelling, virtual organisations and learning will be the sandbox techniques used to build this next generation of software systems.

We envisage a new programming paradigm that is based on two concepts:

1. a Normative context, that determines the rules of the game, i.e. how the interactions between agents are going to happen, and
2. a call-by- agreement interaction method that is based on a two step process:
   - first the establishment of an agreement for action between the agents that respects the normative context, and
   - second, the actual program call for the enactment of the action.

We will also address the need for software engineering methodologies that deal with the issues raised in the project.
The development of agent technologies has taken place within a context of wider visions for information technology.

There are also several key trends and drivers that suggest that agents and agent technologies will be vital. The four considered below are examples; for a large list, see:

- Semantic Web
- Web Services and Service Oriented Computing
- Grid Computing
- Ambient Intelligence
Agreement Technologies
Agreement Technologies

Critical aspects:

- **autonomy and interaction**

- At the agent-level, agents must establish commitments to bringing about goals in the context of more general autonomous behaviour, typically seeking to maximize their utility.

- To constrain the potential excesses of autonomous behaviour, agreements are made between agents, at the interaction-level, providing some kind of guarantee supporting inter-agent relationships.

- Such agreements may be informal commitments between individual agents or services, or they may be stronger contractual commitments between individuals or organizations, with enforcement and/or penalties included as part.

- Techniques that enable software components to reach agreements typically on the performance of services.
Agent and Agreement Technology Challenges

- Trust and reputation.
- Virtual organisation formation and management.
- Resource allocation and coordination.
- Negotiation and Argumentation.
- Contracting and Verification.
- Methodologies.
- Service architecture and composition.
How do agents reaching agreements?

- In an extreme case no agreement is possible — but in most scenarios, there is potential for mutually beneficial agreement on matters of common interest.

- The capabilities of **negotiation and argumentation** are central to the ability of an agent to reach such agreements.

- Negotiation is governed by a particular mechanism, or protocol, this mechanism defines the “rules of encounter” (step by step) between agents (auctions, bargaining).
Negotiation

- Auctions are negotiation mechanisms, but they are only concerned with the allocation of goods: richer techniques for reaching agreements are required.

- Negotiation is the process of reaching agreements on matters of common interest.

- Any negotiation setting will have three components:
  - A negotiation set: possible proposals that agents can make.
  - A protocol.
  - Strategies, one for each agent, which are private. A rule that determines when a deal has been struck and what the agreement deal is. Negotiation usually proceeds in a series of rounds, with every agent making a proposal at every round.
Argumentation

- Argumentation is the process of attempting to convince others of something. Gilbert (1994) identified 4 modes of argument:

  1. **Logical mode.** “If you accept that A and that A implies B, then you must accept that B”.

  2. **Emotional mode.** “How would you feel if it happened to you?”

  3. **Visceral mode.** “Cretin!”

  4. **Kisceral mode.** “This is against Christian teaching!”
Norms

- Heterogeneous and autonomous agents with undesirable and self-interested behaviours -> regulation are required.

- Coordination mechanism.

- Define constraints on agent functionality:
  - Who is authorized to request a service?
  - When and How a service can be provided?

- Normative Language Controlling Access to Services

- Implementation of the Norms Rule-based System
Trust and Reputation

- Computational trust and reputation mechanisms for virtual societies is a recent discipline oriented to increase the reliability and performance of electronic communities by introducing in such communities these well known human social control mechanisms.

- Computational trust and reputation systems have been recognized as key factors for successful electronic commerce adoption.

- These systems are used by intelligent software agents both as a mechanism of search for trustworthy exchange partners and as an incentive in decision-making about whether or not to honour contracts.

- Reputation is also used in electronic environments as a trust-enforcing, deterrent, and incentive mechanism to avoid cheaters and frauds.
We need a framework!!!!!

MAS

- Intelligent and Social Capabilities
- Organizational Concepts
- Normative Concepts
- Negotiation
- Argumentation
- Trust and Reputation

SERVICES

- Standards
- Reuse and Adaptability
- Infrastructure
- Loosely-coupled Distributed Systems

THOMAS
Thomas Architecture

**Platform Kernel**

*AMS*
Agent live cycle control

*Network Layer*
Communication Control

Diagram showing the relationship between OMS, SF, AMS, and Network Layer.
Thomas Architecture

**Organization Management System**
Organization live cycle
Organization dynamics

Diagram:
- OMS
- SF
- AMS
- Network Layer
- Platform Kernel (PK)

17/09/2010
Thomas Architecture

Service Facilitator
- Discovering
- Composition
- Matchmaking

Platform Kernel (PK)

OMS

AMS

Network Layer
Thomas Architecture

Organization Execution Framework

OMS

AMS

Network Layer

Platform Kernel (PK)

SF
THOMAS FRAMEWORK

Service
Agent
Organization

Ag1

Ag2

Ag3

Org1

OMS

SF

THOMAS Framework

SPARQL

WSDL

OWL-S

OWL

SWELL

MINDSWAP
SERVICE IMPLEMENTATION

Apache Tomcat - Axis 2

serviceProfile.owl

serviceProcess.owl

service.war

OWL-S
SF IMPLEMENTATION

FIPA Request Protocol

Magentix Agent

API OWL-S Mindswap

Sparql

MySQL database JENA

Apache Tomcat - Axis 2

OWLS-S

war war war war war

SF Services

THOMAS Framework

OWL-S
OMS IMPLEMENTATION
Thomas Framework

http://users.dsic.upv.es/grupos/ia/sma/tools/Thomas/index.html
Thomas Framework

• Platform independent.
  Thomas framework has been implemented using as kernel platform both MAGENTIX and JADE.

• Gives support to Virtual Organizations.

• Functionalities are described and provided as semantic services.

• Allows the adaptation of the structure and functionality.

• Provides discovering and composition services.
To conclude ……

- Despite the benefits, agent and agreement technologies have not yet entered the mainstream in the way that object-oriented technologies have.

- The majority of commercial organisations adopting agent technologies would be classified as early adopters, so considerable potential exists for further applications of the technology.

- To date, the range of applications has included:
  - automated trading in online marketplaces;
  - simulation and training applications in defence domains;
  - network management in utilities networks;
  - user-interface and local interaction management in telecommunication networks;
  - schedule planning and optimisation in logistics and supply-chain management;
  - control system management in industrial plants, such as steel works; and
  - simulation modelling to guide decision-makers in public policy domains, such as transport and medicine.
Welcome to Agreement-Technologies — Agreement Technologies

Agreement Technologies

Agreement is one of the crucial social concepts that helps human agents cope with their social environment and is present in all human interactions. In fact, without agreement there is no cooperation and ultimately social systems cannot emerge. Agreement is necessary in our everyday life.

Until recently, the concept of agreement was a domain of study mainly for philosophers, sociologists and was only applicable only to human societies. In the last thirty years, the growth of disciplines such as social psychology, sociology, social neuroscience, and the development of adaptive multi-agent artificial systems (MAS), together with the spectacular emergence of the information society technologies, have changed this situation. Presently, agreement and all the processes and mechanisms implicated in reaching agreements between different kinds of agents are a subject of research and analysis with very many perspectives.

Provided that all these approaches are relevant for a robust understanding and an efficient implementation of artificial social systems, the dialogue and knowledge transfer among these new disciplines appears as a must in social-oriented science and technology.
http://www.agreement-technologies.eu/

The Action

Agreement Technologies (AT) refer to computer systems in which autonomous software agents negotiate with one another, typically on behalf of humans, in order to come to mutually acceptable agreements.

This Action aims at coordinating national efforts on a new paradigm for next generation distributed systems, based on the concept of agreement between computational agents. An entity may choose whether to fulfill an agreement or not, and it should fulfill it when there is an obligation to do so derived from the standing agreements. Autonomy, interaction, mobility and openness are the characteristics that the paradigm will cover from a theoretical and practical perspective. Semantic alignment, negotiation, argumentation, virtual organisations, learning, real time, and several other technologies will be in the sandbox to define, specify and verify such systems. Both functional and non-functional properties are to be studied. Security on execution will be based on trust and reputation measures. These measures will help agents to determine with whom to interact and what terms and conditions to accept.
AGENTS AND AGREEMENT TECHNOLOGIES: THE NEXT GENERATION OF DISTRIBUTED SYSTEMS

Vicent J. Botti Navarro
Grupo de Tecnología Informática- Inteligencia Artificial
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia