Architecture of software systems

Course 12: Intelligent systems, Multi-agent systems

David Šišlák
david.sislak@fel.cvut.cz

Intelligent systems

- intelligent system integrates concepts of **artificial intelligence** such as:
  - knowledge reprepresentation and expert systems
  - advanced search methods
  - mathematical reasoning methods
  - **nature inspired computing**: artificial neural networks and genetic algorithms
  - **agent-based computing**: distributed artificial intelligence and multi-agent systems

Introduction to agents #1

» **autonomous agents and multi-agent systems** (also referred to as agent-based computing):
  - a specific sub-field of computer science and artificial intelligence,
  - investigates the concepts of autonomous decision making, communication and coordination, distributed planning and distributed learning but also game-theoretic aspects of competitive behavior or logical formalization of higher level knowledge structures representing interaction attitude of actors in multi-actor environment.

A multi-agent system is a decentralized computational (software) system, often distributed (or at least open to distribution across hardware platforms) whose behavior is defined and implemented by means of complex, peer-to-peer interaction among autonomous, rational and deliberative units – agents.

Introduction to agents #2

An **agent** is an encapsulated computational (or physical, even human) system, that is ***situated in*** some ***environment***, and that is capable of flexible, ***autonomous behavior*** in order to meet its design objective (Wooldridge, 2000). The agent can exists on its own but often is a component of a ***multi-agent system***.

Agent technology provides a set of ***tools***, **algorithms** and **methodologies** for development of distributed, asynchronous intelligent software applications that leverage the above listed theories.

Key properties of an autonomous intelligent agent

- **Autonomy** – the agent is accountable for execution of its own actions and is not controlled from outside. Often the agent's reasoning mechanism that selects the action to be executed is unknown from outside of the agent (unlike e.g. objects).
- **Reactivity** – the agent is able react qucikly to the events in the environment and to the requests from other agents, it is able to reconsider its activity according to the change of the environment in timely fashion. Often the longest reasoning cycle of an agent needs to perform faster than the fastest change in the environment (calculative rationality).
- **Intentionality** – the agents is able to maintain its long term intention encoded by the agent's designer and is capable to consider both the long term intentions and immediate reactive inputs when selecting the next action.
- **Social capability** – the agent is able to interact, collaborate, form teams but also to perform different levels of reasoning about the other agents.

Agent design levels

Agents design levels:

» **agent-level**: concern individual agents (agent architecture, reasoning, learning, local processing of social knowledge);

» **interaction-level**: concern communication among agents (languages, interaction protocols, negotiations, resource allocation mechanisms);

» **organization-level**: related to the agent communities as a whole (organizational structure, trust, norms, obligations, self-organization, etc.).

Similar concepts

objects - computational entity with its ***encapsulated state***, ability to perform methods on the state and communicating with the other objects e.g. via message passing

- lesser **degree of autonomy** - possibility to have a public method
- joint goal is set-up at the **design-time**
- multi-agent systems are inherently **multi-threaded**

expert systems - the most important technology of the 1980's

- expert systems are **disembodied** from the environment
- expert systems are not capable of **reactive** and **proactive behavior**
- expert systems are not equipped with the **social ability**

Agent technology domains

- **manufacturing**: planning highly complex production, control of dynamic, unpredictable, unstable processes, diagnostics, repair, reconfiguration/replanning.
- **virtual enterprises**: forming business alliances, forming long-term/short-term deals, managing supply chains.
- **internet agents**: mainly for intelligent shopping and auctioning, information retrieval and searching, remote access to information and remote system control.
- **transport**: intelligent car, public transport, logistic and material handling, but also peace-keeping missions, military maneuvers, etc.
- **collective robotics operations**: cooperation and autonomy in the group of robotic entities (UAS, ground vehicles, unattended sensors), replacement of teleoperation with autonomous decision making
- **utility networks**: energy distribution networks, mobile operators networks, cable provider networks - simulation and predication of alarm situations, prevention to black-out and overload, intrusion detection.

Agent architectures #1

- **Reactive agents** are agents that contain no symbolic knowledge representation (ie: no state, no representation of the environment, no representation of the other agents, ...). Their behaviour is defined by a set of perception-action rules.

$$\text{rules} \times \text{percept} \rightarrow \text{action}$$

Agent architectures #2

The classical approach to building agents is to view them as a particular type of knowledge-based system, and bring all the associated methodologies of such systems to bear. We define a **deliberative agent** architecture to be one that:

- contains an explicitly represented, **symbolic model of the world**;
- makes decisions (e.g. about what actions to perform) e.g. via **symbolic reasoning**.

Agent architectures #3

Belief-desire-intention (BDI) model is framework for reasoning about formal abstract models of mental states (based on Theory of Practical Reasoning).

» contains representations (as objects, data structures, or whatever) of:

- **beliefs**, which constitute its knowledge of the state of its environment (and perhaps also some internal state),
- **desires**, which determine its motivation what it is trying to bring about, maintain, find out, etc.,
- **intentions**, which capture its decisions about how to act in order to fulfill its desires (committed desires)

» **intention** is something between the agents' **state of mind** (belief) and the immediate action to be performed

» unlike **desire/goal** an intention may be seen as agents immediate **commitment** to implementing an action.

Interaction schemes

From the point of view of interaction among agents we distinguish:

» **Open systems**

- Interaction among various types of agents from different developers
- Common understanding of messages is necessary – specification of message structure
- May act as self-interested and try to harm others/whole system
  - security issues
- Strong emphasis on interoperability

» **Closed systems**

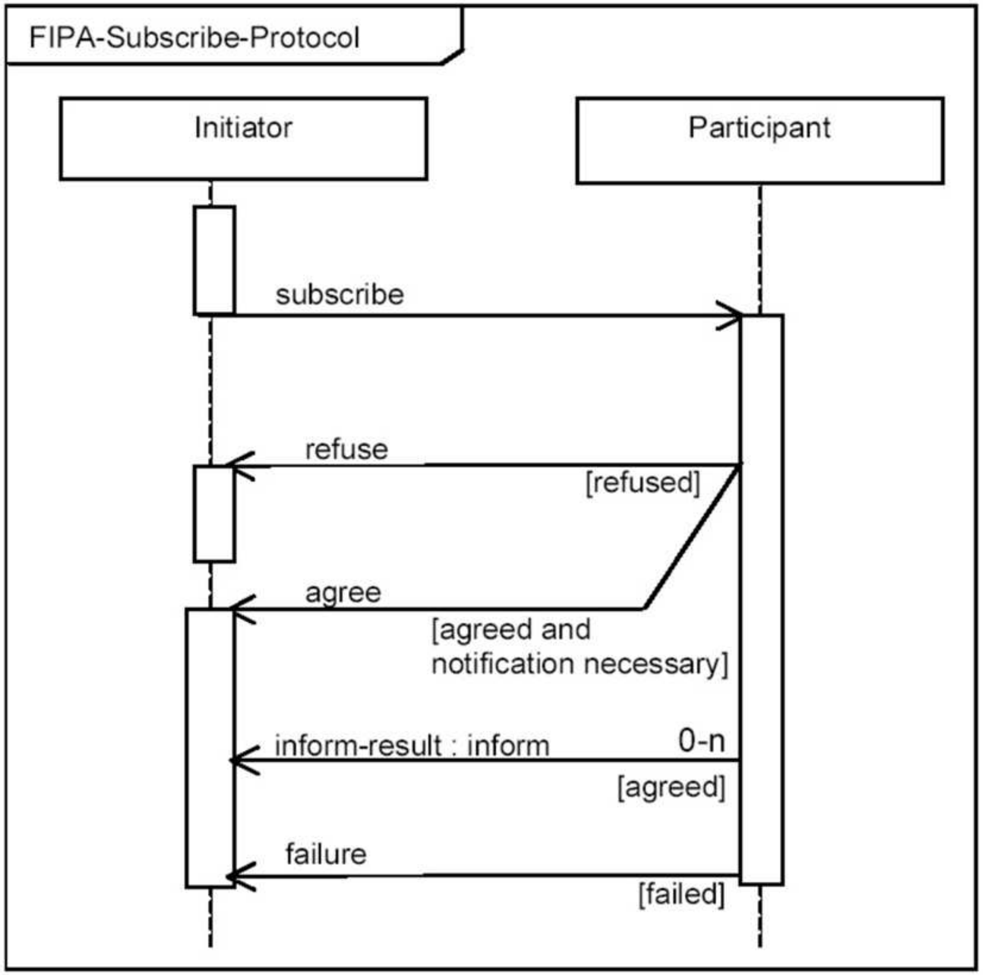
- Interact with a predefined set of agents known to the developer in advance
- Proprietary data formats can be used
- Interoperability can be sacrificed for the sake of performance optimizations

FIPA specifications #1

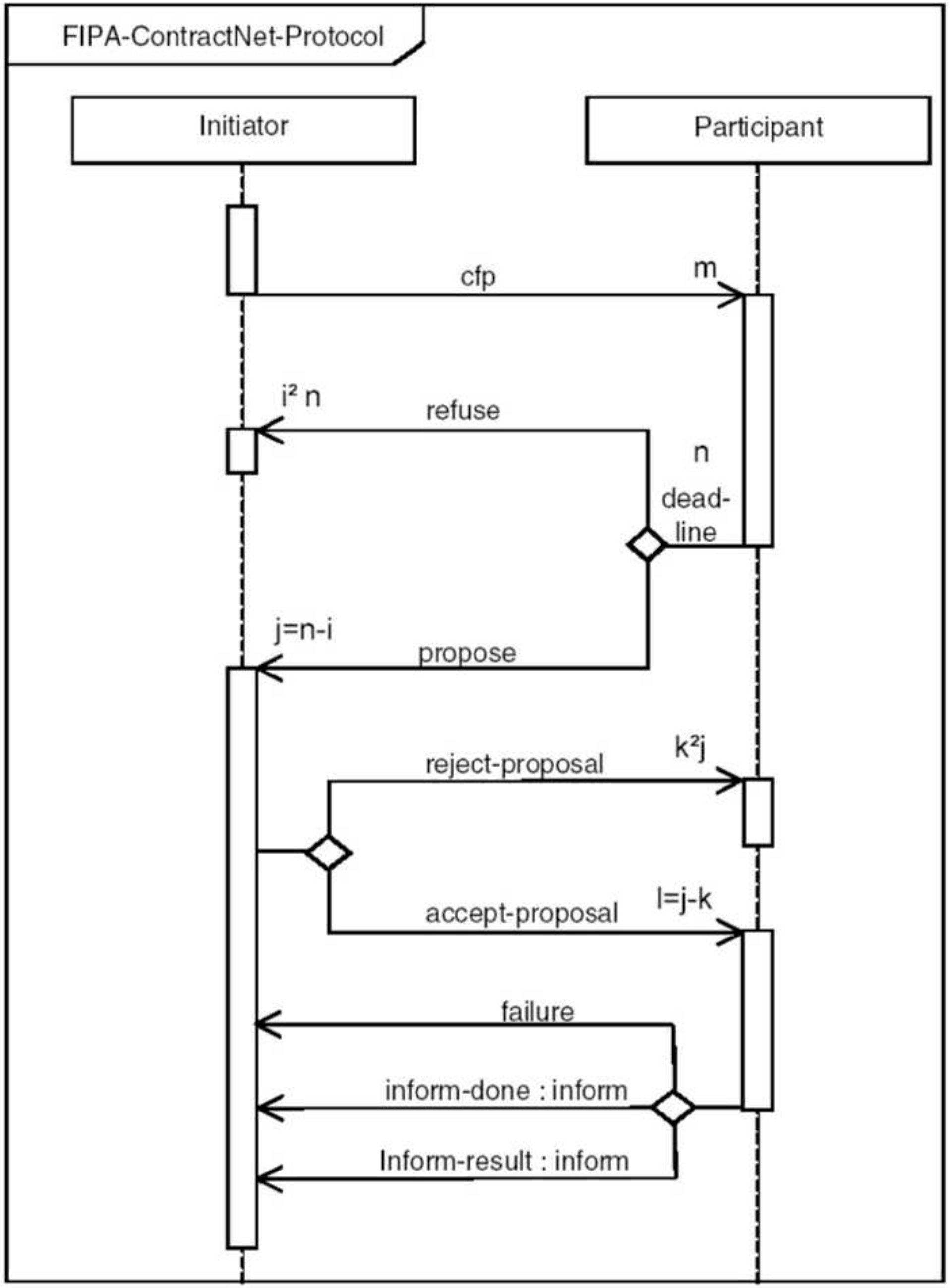
Foundation for Intelligent Physical Agents - FIPA

- Founded to create specification that will ***ensure interoperability among agents***
- Complete set of ***specifications*** from different categories:
  - agent communication
  - agent transport
  - agent management
  - abstract architecture and applications
- Most significant for agent interoperability is agent communication and transport

Agent communication language – Subscribe-advertise



Agent communication language - CNP



Mobile agents #1

Mobile agents are characterized by code mobility

- Mobile agents travel to places, where they perform tasks on behalf of a user
- Reason why to travel - insufficient computational power, unreliable communication, remote data source
- Security issues
- Mobility vs. cloning, stand-in agents

Types of agent mobility

- **Strong mobility** - mobility of code, data and execution state, transparent for the computational process, requires support from the OS and execution environment
- **Weak mobility** - only the code and data are transferred, intentional mobility

Mobile agents #2

Problems with agent mobility

» Various operating systems, programming languages, agent platforms

» Security features - permissions, authorities, access control

» Necessity to transfer all required data, libraries

» Interaction with message transport system - new communication address, delivery of messages received by old message transport

Security

Important aspect especially in the case of open systems

- **Thread-safe agent execution** model which holds rest of the system harmless against agent failure
- **Communication security**
  - message encryption/signing
  - security certificates with public/private keys
- **Trust and reputation** models to create a set of trustful collaborators in open systems
- **Protect private knowledge** of individual agents