EVROPSKÁ UNIE

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

OP Vzdělávání pro konkurenceschopnost

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Introduction to agent-based computing

doc. Dr. Ing. Michal Pěchouček, M.Sc.

March 22, 2010

Tato prezentace je spolufinancována Evropským sociálním fondem a státním rozpočtemČeské republiky.

Intelligent systems
Recent trends in intelligent systems
Introduction to agents
Agent architectures
Formal model of agency

Intelligent systems

Recent trends in intelligent systems

An **intelligent** system is a computational (=software) system that is capable of intelligent decision making. An example of such decision making is:

- planning and scheduling,
- system diagnostics,
- natural language understanding, language-to-language translation
- intelligent control, exploration (e.g. in robotics)
- game planning, problem solving and theorem proving

Intelligent systems

Intelligent system integrates concepts of **artificial intelligence** such as:

- knowledge representation 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

Motivation/vision thing

in computer science and artificial intelligence there are 5 important trends:

- **ubiquity:** computation, internet access, agents to be everywhere
- **interconnection:** computation to be distributed, networked
- **intelligence:** in order to master very complex problems
- **delegation:** computers will be given control to imitate humans decisions; **human orientation**: adapt to the way how we do reasoning
- how can **cooperation emerge** in societies of self-interested agents?
- what **language** can the agents use to communicate their beliefs and aspiration both to **agents** and to **people**?
- how can the agents **recognized** that their beliefs are in **conflict**?
- how can the **self-interested agents reach an agreement** with one another?
- how can they **coordinate activities** so that they achieve the goal **collaboratively**?

Intelligent systems
Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains
Agent architectures
Formal model of agency

Introduction to agents

Intelligent systems

Introduction to agents

Introduction to agents

Agent concepts

Similar concepts

Research challenges and applications of agent based computing

Agent applications

Problem domains

Agent architectures

Formal model of agency

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 behaviour 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

Intelligent systems
Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains
Agent architectures
Formal model of agency

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

Intelligent systems
Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains
Agent architectures
Formal model of agency

- **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 her 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 [1].
- **Intentionality** – the agents is able to maintain her 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.

[1] Property referred to as **calculative rationality**

Intelligent systems

Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains

Agent architectures

Formal model of agency

Agents use models:

- **agents as design metaphore**, providing the designers and developers with a way of structuring an application around autonomous, communicative elements;
- **source of technologies**, providing specific techniques and algorithms for dealing with interactions in dynamic and open environments and
- **simulation models**, providing strong models for representing complex and dynamic real-world environments.

Agents design levels:

- **organization-level**: related to the agent communities as a whole (organizational structure, trust, norms, obligations, self-organisation, etc.);
- **interaction-level**: concern communication among agents (languages, interaction protocols, negotiations, resource allocation mechanisms);
- **agent-level**: concern individual agents (agent architecture, reasoning, learning, local processing of social knowledge).

Similar concepts

Intelligent systems

Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains

Agent architectures

Formal model of agency

objects - computational entity with its encapsulated state, ability to perform methods on the state and communicating with the other objects 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**

Research challenges and applications of agent based computing

Intelligent systems
Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains
Agent architectures
Formal model of agency

- **Coordination** - list of agent techniques (based mainly on dedicated coordination protocols and various collaboration enforcement mechanisms) that facilitates coordinated behavior between autonomous, while collaborative agents. Coordination usually supports conflict resolution and collision avoidance, resource sharing, plan merging, and various collective kinds of behavior.
- **Negotiation** - list of various negotiation and auctioning techniques that facilitate an agreement about a joint decision among several self-interested actors or agents. Here we emphasize mainly negotiation protocols and mechanisms how individual actors shall act and what strategies shall they impose to optimize their individual utility.

Intelligent systems

Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains

Agent architectures

Formal model of agency

- **Simulation** - techniques that allow inspection of collective behavior of the interactive actors, provided that the models of the individual agents are known. Here we count on the versatile simulation frameworks that allow long-run complex simulation and various "what-if" analyses of different problems. If distributed hardware system is modeled, agent-based simulation enables a close linkage between simulation and the real hardware machinery.

- **Interoperability** - set of techniques for achieving high level interoperability among software components developed by different designers, especially in the situation where the source code and complete models of behavior are not shared. Interoperability is studied on the level of physical connections via interaction protocols but also semantics of communication.

Intelligent systems

Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains

Agent architectures

Formal model of agency

- **Adjustable Autonomy and Policies** - set of techniques and methods for specifying and dynamic adjustment of decision making autonomy of the individual actors in a multi-agent system. Various formal frameworks for specifying policies have been proposed and numerous policy management systems have been designed by the agent community.

- **Organization** - techniques supporting agents in ability to organize autonomously in permanent or temporal interaction and collaboration structures (virtual organizations), assign roles, establish and follow norms, or comply with electronic institutions.

Intelligent systems
Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains
Agent architectures
Formal model of agency

- **Multi-agent Learning** - in the multi-agent community there are various methods allowing an agent to form hypothesis about available agents. These methods work mainly with the logs of communication or past behavior of agents. Agent community also provides techniques for collaborative (distributed) learning, where agents may share learnt hypothesis or observed data. A typical application domain is distributed diagnostics.
- **Multi-agent Planning** - specific methods of collaboration and sharing information while planning operation among autonomous collaborating agents. Agent community provides methods for knowledge sharing, negotiation and collaboration during the 5 phases of distributed planning (Durfee, 1999): task decomposition, resource allocation, conflict resolution, individual planning, and plan merging. These methods are particularly suitable for the situations when the knowledge needed for planning is not available centrally.

Intelligent systems
Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains
Agent architectures
Formal model of agency

- **Knowledge Sharing** - techniques assisting in sharing knowledge and understanding different types of knowledge among collaborative parties as well as methods allowing partial knowledge sharing in semi-trusted agent communities (closely linked with distributed learning and distributed planning).
- **Trust and Reputation** - methods allowing each agent to build a trust model and share reputation information about agents. Trust and reputation is used in non-collaborative scenario where agents may perform non-trusted and deceptive behavior.

Intelligent systems
Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains
Agent architectures
Formal model of agency

Distributed and collective aspects :

- **Decentralized scenarios:** Particularly suitable are the domains where the data and knowledge required for computation are not or cannot be available centrally or the process physical system control needs to be distributed.
  - **Geographical distribution** of knowledge and control (e.g., logistics, collaborative exploration, mobile and collective robotics, pervasive systems) or the environments with partial or temporary communication inaccessibility.
  - **Competitive domains**, with the restrictions on the information sharing (e.g., e-commerce applications, supply-chain management, and e-business)
  - Domains with the requirements for **time-critical response** and **high robustness** in distributed environment (e.g., time-critical (soft- and/or hard-realtime) manufacturing or industrial systems control, with re-planning, or fast local reconfiguration)

Agent applications

Intelligent systems
Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains
Agent architectures
Formal model of agency

- **Simulation and modeling scenarios:** Using agents for simulation purposes has been very common, while the right justification was often missing. Agents shall be deployed in simulation exercises where we require, e.g., an easy migration from the simulation to deployment in real environment.
- **Open systems scenarios:** In scenarios requiring integration and interoperability among software systems that are not known a priori and whose source code may not be available - here the use of agent technologies, especially agent communication languages and interoperability standards is advisable.
- **Complex systems scenarios:** In scenarios requiring modeling, controlling or engineering of complex systems. Decomposition of the decision making into separate agents' reasoning and solving problems by means of negotiation represents a novel software development paradigm .

Intelligent systems

Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains

Agent architectures

Formal model of agency

Autonomy oriented aspects of agency is appropriate in application domains with high requirements for systems with decision making autonomy, when the user delegates the substantial amount of decision making authority to the system and when the system is expected to cope independently with unexpected situations.

Problem domains

Intelligent systems
Introduction to agents
Introduction to agents
Agent concepts
Similar concepts
Research challenges and applications of agent based computing
Agent applications
Problem domains
Agent architectures
Formal model of agency

- **manufacturing:** planning highly complex production, control of dynamic, unpredictable, unstable processes, diagnostics, repair, reconfiguration/replanning.
- **virtual enterprizes:** 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.

Intelligent systems
Introduction to agents
Agent architectures
Reactive agents
Deliberative agents
BDI Agents
Formal model of agency

Agent architectures

Intelligent systems
Introduction to agents
Agent architectures
Reactive agents
Deliberative agents
BDI Agents
Formal model of agency

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.

$$\wp(\texttt{rules}) \times \texttt{percept} \rightarrow \wp(\texttt{action})$$

Deliberative agents

Intelligent systems
Introduction to agents
Agent architectures
Reactive agents
Deliberative agents
BDI Agents
Formal model of agency

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

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

BDI Agents

Intelligent systems
Introduction to agents
Agent architectures
Reactive agents
Deliberative agents
BDI Agents
Formal model of agency

Beleif-desire-intention 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.

BDI examples

Intelligent systems
Introduction to agents
Agent architectures
Reactive agents
Deliberative agents
BDI Agents
Formal model of agency

BDI examples

Intelligent systems
Introduction to agents
Agent architectures
Reactive agents
Deliberative agents
BDI Agents
Formal model of agency

- (Bel john (sunny melbourne))

BDI examples

Intelligent systems
Introduction to agents
Agent architectures
Reactive agents
Deliberative agents
BDI Agents
Formal model of agency

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))

BDI examples

Intelligent systems
Introduction to agents
Agent architectures
Reactive agents
Deliberative agents
BDI Agents
Formal model of agency

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))
- $\forall A$ (english-person A) $\Rightarrow$ (Bel A (sunny melbourne))

BDI examples

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))
- $\forall A$ (english-person A) $\Rightarrow$ (Bel A (sunny melbourne))
- $\forall x$ (australian-city x) $\Rightarrow$ (Bel john (sunny x))

BDI examples

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))
- $\forall A$ (english-person A) $\Rightarrow$ (Bel A (sunny melbourne))
- $\forall x$ (australian-city x) $\Rightarrow$ (Bel john (sunny x))
- $\forall x$ (Bel john x) $\Rightarrow$ (Bel jim x)

BDI examples

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))
- $\forall A$ (english-person A) $\Rightarrow$ (Bel A (sunny melbourne))
- $\forall x$ (australian-city x) $\Rightarrow$ (Bel john (sunny x))
- $\forall x$ (Bel john x) $\Rightarrow$ (Bel jim x)
- (Int author (write book))

BDI examples

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))
- $\forall A$ (english-person A) $\Rightarrow$ (Bel A (sunny melbourne))
- $\forall x$ (australian-city x) $\Rightarrow$ (Bel john (sunny x))
- $\forall x$ (Bel john x) $\Rightarrow$ (Bel jim x)
- (Int author (write book))
- (Des author ($\forall A$ (Int A (buy book)))

BDI examples

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))
- $\forall A$ (english-person A) $\Rightarrow$ (Bel A (sunny melbourne))
- $\forall x$ (australian-city x) $\Rightarrow$ (Bel john (sunny x))
- $\forall x$ (Bel john x) $\Rightarrow$ (Bel jim x)
- (Int author (write book))
- (Des author ($\forall A$ (Int A (buy book)))
- (Des A (AF (win A))) $\wedge$ (Int A (EF (buy-ticket A))) $\wedge\neg$ (Bel A (AF (win A)))

BDI examples

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))
- $\forall A$ (english-person A) $\Rightarrow$ (Bel A (sunny melbourne))
- $\forall x$ (australian-city x) $\Rightarrow$ (Bel john (sunny x))
- $\forall x$ (Bel john x) $\Rightarrow$ (Bel jim x)
- (Int author (write book))
- (Des author ($\forall A$ (Int A (buy book)))
- (Des A (AF (win A))) $\wedge$ (Int A (EF (buy-ticket A))) $\wedge\neg$ (Bel A (AF (win A)))
- $\forall A, B$ (motherof A B) $\Rightarrow$ (Bel B (wonderfull A))

BDI examples

Intelligent systems
Introduction to agents
Agent architectures
Reactive agents
Deliberative agents
BDI Agents
Formal model of agency

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))
- $\forall A$ (english-person A) $\Rightarrow$ (Bel A (sunny melbourne))
- $\forall x$ (australian-city x) $\Rightarrow$ (Bel john (sunny x))
- $\forall x$ (Bel john x) $\Rightarrow$ (Bel jim x)
- (Int author (write book))
- (Des author ($\forall A$ (Int A (buy book)))
- (Des A (AF (win A))) $\wedge$ (Int A (EF (buy-ticket A))) $\wedge\neg$ (Bel A (AF (win A)))
- $\forall A, B$ (motherof A B) $\Rightarrow$ (Bel B (wonderfull A))
- (Des janine $\forall A$ (Bel A (dangerous ozonehole)))

BDI examples

- (Bel john (sunny melbourne))
- $\exists A$ (Bel A (sunny melbourne))
- $\forall A$ (english-person A) $\Rightarrow$ (Bel A (sunny melbourne))
- $\forall x$ (australian-city x) $\Rightarrow$ (Bel john (sunny x))
- $\forall x$ (Bel john x) $\Rightarrow$ (Bel jim x)
- (Int author (write book))
- (Des author ($\forall A$ (Int A (buy book)))
- (Des A (AF (win A))) $\wedge$ (Int A (EF (buy-ticket A))) $\wedge \neg$ (Bel A (AF (win A)))
- $\forall A, B$ (motherof A B) $\Rightarrow$ (Bel B (wonderfull A))
- (Des janine $\forall A$ (Bel A (dangerous ozonehole)))
- (Bel Tony $\forall B$ (Bel B (Int Tony $\neg$(prime-minister Tony))))

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Formal model of agency

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Agent is an encapsulated computational system, that is **situated in some environment**, and that is capable of flexible, autonomous behaviour in order to meet its design objective (Wooldridge).
an agent has got a **partial control** over the environment – is nondeterministic

Environment is

- accessible vs. inaccessible
- dynamic vs. static
- deterministic vs. non-deterministic
- discrete vs. continuous

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Let us have

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Let us have

- possible states of the environment: $S = \{s, s', \dots\}$,

Agents abstract architecture

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Let us have

- possible states of the environment: $S = \{s, s', \dots\}$,
- possible actions that can transform the environment: $\mathcal{A}c = \{\alpha, \alpha', \dots\}$

Agents abstract architecture

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Let us have

- possible states of the environment: $S = \{s, s', \dots\}$,
- possible actions that can transform the environment: $\mathcal{A}c = \{\alpha, \alpha', \dots\}$
- simple model of the environment: $\tau : S \times \mathcal{A}c \to \wp(S)$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Let us have

- possible states of the environment: $S = \{s, s', \dots\}$,
- possible actions that can transform the environment: $\mathcal{A}c = \{\alpha, \alpha', \dots\}$
- simple model of the environment: $\tau : S \times \mathcal{A}c \to \wp(S)$
- **stateless** purely, reactive agents: $\mathcal{A}g : S \to \mathcal{A}c$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Agents abstract architecture

Let us have

- possible states of the environment: $S = \{s, s', \dots\}$,
- possible actions that can transform the environment: $\mathcal{A}c = \{\alpha, \alpha', \dots\}$
- simple model of the environment: $\tau : S \times \mathcal{A}c \to \wp(S)$
- **stateless** purely, reactive agents: $\mathcal{A}g : S \to \mathcal{A}c$
- **stateful** agents: $\mathcal{A}g : S^* \to \mathcal{A}c$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Agents abstract architecture

Let us have

- possible states of the environment: $S = \{s, s', \dots\}$,
- possible actions that can transform the environment: $\mathcal{A}c = \{\alpha, \alpha', \dots\}$
- simple model of the environment: $\tau : S \times \mathcal{A}c \to \wp(S)$
- **stateless** purely, reactive agents: $\mathcal{A}g : S \to \mathcal{A}c$
- **stateful** agents: $\mathcal{A}g : S^* \to \mathcal{A}c$
- **goal-directed**, **knowledgeable** agents: $\mathcal{A}g : B \times G \to \mathcal{A}c$

Agents abstract architecture

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Let us have

- possible states of the environment: $S = \{s, s', \dots\}$,
- possible actions that can transform the environment: $\mathcal{A}c = \{\alpha, \alpha', \dots\}$
- simple model of the environment: $\tau : S \times \mathcal{A}c \rightarrow \wp(S)$
- **stateless** purely, reactive agents: $\mathcal{A}g : S \rightarrow \mathcal{A}c$
- **stateful** agents: $\mathcal{A}g : S^* \rightarrow \mathcal{A}c$
- **goal-directed**, **knowledgeable** agents: $\mathcal{A}g : B \times G \rightarrow \mathcal{A}c$
- **autonomous** agents: $\mathcal{A}g : S^* \rightarrow \wp(\mathcal{A}c)$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

given the **model of the history** of an agent operation A in the environment

$$r : \{s_0 \xrightarrow[\alpha_0]{} s_1 \xrightarrow[\alpha_1]{} s_2 \xrightarrow[\alpha_2]{} \dots \xrightarrow[\alpha_3]{} s_n\}$$

we introduce $\mathcal{R}$ – **characteristic behavior** – as the set of all finite possible histories

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

environment is either

- deterministic, history independent: $\tau : S \times Ag \rightarrow S$
- non-deterministic, history independent: $\tau : S \times Ag \rightarrow \wp(S)$
- history dependent environment $\mathcal{E}_{nv} = \langle S, s_0, \tau \rangle$
  - $\mathcal{R}^{\mathcal{A}c} \subseteq \mathcal{R}$ so that it finishes with an action,
  - $\mathcal{R}^{\mathcal{E}_{nv}} \subseteq \mathcal{R}$ so that it finishes with an environment state

$$\tau : \mathcal{R}^{\mathcal{A}c} \rightarrow \wp(S)$$

- model of agents that inhabit the system:

$$\mathcal{A}g : \mathcal{R}^{\mathcal{E}_{nv}} \rightarrow \wp(\mathcal{A}c)$$

Agents abstract architecture

characteristic behavior $\mathcal{R}$ of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ is $\mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv})$
if we consider only **finite histories**, the sequence $r : \{s_0 \overrightarrow{\alpha_0} s_1 \overrightarrow{\alpha_1} s_2 \overrightarrow{\alpha_2} \dots \overrightarrow{\alpha_3} s_n\}$ represents a behavior of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ provided that:

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

characteristic behavior $\mathcal{R}$ of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ is $\mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv})$

if we consider only **finite histories**, the sequence $r : \{s_0 \overrightarrow{\alpha_0} s_1 \overrightarrow{\alpha_1} s_2 \overrightarrow{\alpha_2} \dots \overrightarrow{\alpha_3} s_n\}$ represents a behavior of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ provided that:

- s_0 is an initial state of $\mathcal{E}_{nv}$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

characteristic behavior $\mathcal{R}$ of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ is $\mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv})$
if we consider only **finite histories**, the sequence $r : \{s_0 \overrightarrow{\alpha_0} s_1 \overrightarrow{\alpha_1} s_2 \overrightarrow{\alpha_2} \dots \overrightarrow{\alpha_3} s_n\}$ represents a behavior of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ provided that:

- s_0 is an initial state of $\mathcal{E}_{nv}$
- $\alpha_0 \in \mathcal{A}g(s_0)$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

characteristic behavior $\mathcal{R}$ of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ is $\mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv})$
if we consider only **finite histories**, the sequence $r : \{s_0 \xrightarrow[\alpha_0]{} s_1 \xrightarrow[\alpha_1]{} s_2 \xrightarrow[\alpha_2]{} \dots \xrightarrow[\alpha_3]{} s_n\}$ represents a behavior of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ provided that:

- s_0 is an initial state of $\mathcal{E}_{nv}$
- $\alpha_0 \in \mathcal{A}g(s_0)$
- $\forall n > 0 :$

Intelligent systems

Introduction to agents

Agent architectures

Formal model of agency

Agent and the environment

Agents abstract architecture

Agents' utility function

Rational agents' expected utility

Agents' bounded and calculative rationality

Task specification

characteristic behavior $\mathcal{R}$ of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ is $\mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv})$

if we consider only **finite histories**, the sequence $r : \{s_0 \xrightarrow[\alpha_0]{} s_1 \xrightarrow[\alpha_1]{} s_2 \xrightarrow[\alpha_2]{} \ldots \xrightarrow[\alpha_3]{} s_n\}$ represents a behavior of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ provided that:

- s_0 is an initial state of $\mathcal{E}_{nv}$
- $\alpha_0 \in \mathcal{A}g(s_0)$
- $\forall n > 0 :$

  $s_n \in \tau((s_0, \alpha_0, \ldots s_{n-1}, \alpha_{n-1}))$ and

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

characteristic behavior $\mathcal{R}$ of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ is $\mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv})$

if we consider only **finite histories**, the sequence $r : \{s_0 \xrightarrow[\alpha_0]{} s_1 \xrightarrow[\alpha_1]{} s_2 \xrightarrow[\alpha_2]{} \dots \xrightarrow[\alpha_3]{} s_n\}$ represents a behavior of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ provided that:

- s_0 is an initial state of $\mathcal{E}_{nv}$
- $\alpha_0 \in \mathcal{A}g(s_0)$
- $\forall n > 0 :$

  $s_n \in \tau((s_0, \alpha_0, \dots s_{n-1}, \alpha_{n-1}))$ and

  $\alpha_n = \mathcal{A}g((s_0, \alpha_0, \dots s_{n-1}, \alpha_{n-1}, s_n))$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

characteristic behavior $\mathcal{R}$ of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ is $\mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv})$
if we consider only **finite histories**, the sequence $r : \{s_0 \overrightarrow{\alpha_0} s_1 \overrightarrow{\alpha_1} s_2 \overrightarrow{\alpha_2} \dots \overrightarrow{\alpha_3} s_n\}$ represents a behavior of an agent $\mathcal{A}g$ in an environment $\mathcal{E}_{nv}$ provided that:

- s_0 is an initial state of $\mathcal{E}_{nv}$
- $\alpha_0 \in \mathcal{A}g(s_0)$
- $\forall n > 0 :$

  $s_n \in \tau((s_0, \alpha_0, \dots s_{n-1}, \alpha_{n-1}))$ and

  $\alpha_n = \mathcal{A}g((s_0, \alpha_0, \dots s_{n-1}, \alpha_{n-1}, s_n))$

Two agents are **behaviorally equivalent** iff $\mathcal{R}(\mathcal{A}g_1, \mathcal{E}_{nv}) = \mathcal{R}(\mathcal{A}g_2, \mathcal{E}_{nv})$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

short-term utility function – $u : S \to \mathbb{R}$ (associate utilities with individual states - the task of the agent is then to bring about states that maximize utility)
long term utility function –– $u : \mathcal{R} \to \mathbb{R}$ (assigns a utility not to individual states, but to runs themselves)
probability of a run to happen $P(r|\mathcal{A}g, \mathcal{E}_{nv})$

$$\sum_{\forall r : r \in \mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv})} P(r|\mathcal{A}g, \mathcal{E}_{nv}) = 1$$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

expected utility – a measure representing a probability of a all individual runs to happen and the profit they may bring

$$u(r)P(r|\mathcal{A}g, \mathcal{E}_{nv})$$

Rational agent is trying to optimize its profit and thus to **maximize** its utility expected function

$$\mathcal{A}g_{\texttt{rational}} = \text{arg} \max_{\mathcal{A}g \in \mathcal{AG}} \sum_{\forall r : r \in \mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv})} u(r)P(r|\mathcal{A}g, \mathcal{E}_{nv})$$

Agents' bounded and calculative rationality

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

concept of **bounded rationality** and optimality :

- $\mathcal{AG}_m = \{\mathcal{A}g | \mathcal{A}g \in \mathcal{AG}$ and $\mathcal{A}g$ can be implemented on machine $m\}$
- $\mathcal{AG}_p = \{\mathcal{A}g | \mathcal{A}g \in \mathcal{AG}$ and $\mathcal{A}g$ can be implemented so that for any input it gives polynomially bounded output$\}$

  concept of **calculative rationality**

  - $\mathcal{AG}_{CR} = \{\mathcal{A}g | \mathcal{A}g \in \mathcal{AG}$ and $\mathcal{A}g$ can be implemented so that for any reaction in the environment it provides an output faster than that the environment changes again$\}$

Task specification

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

predicate task specification – $\Psi : \mathcal{R} \rightarrow \{0, 1\}$
given a task environment $\langle \mathcal{E}_{nv}, \Psi \rangle$ we say that:

$$\mathcal{R}_\Psi(\mathcal{A}g, \mathcal{E}_{nv}) = \{r | r \in \mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv}) \wedge \Psi(r)\}$$

$$P(\Psi | \mathcal{A}g, \mathcal{E}_{nv}) = \sum_{r = \mathcal{R}_\Psi(\mathcal{A}g, \mathcal{E}_{nv})} P(r | \mathcal{A}g, \mathcal{E}_{nv})$$

Intelligent systems
Introduction to agents
Agent architectures
Formal model of agency
Agent and the environment
Agents abstract architecture
Agents' utility function
Rational agents' expected utility
Agents' bounded and calculative rationality
Task specification

Task specification

success in task completion

- **pessimistic definition** $-\forall r \in \mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv}) : \Psi(r)$
- **optimistic definition** $-\exists r \in \mathcal{R}(\mathcal{A}g, \mathcal{E}_{nv}) : \Psi(r)$

We distinguish between two types of tasks: **maintenance task** and **achievement task**