Agent Theory

Synonyms

Agent Systems, Agent-based Computing, Autonomous Agents, Multi-Agent System

Definition

Agent theory provides the basis for a novel paradigm of computation. While agent-based computing (or simply, agent system) has several roots as far as its concepts, models and enabling technologies are concerned, there is a general consensus about its two main abstractions (Wooldridge 2000, Luck et al. 2006, Shoham and Leyton-Brown 2009):

- An **agent** is a computational system that is situated in an unpredictable, dynamic environment where it is capable of exhibiting autonomous and intelligent behavior.
- An agent's environment typically includes also other agents with diverging information and/or interests. The community of interacting agents, as a whole, operates as a **multi-agent system** (MAS) that can solve such complex problems that are beyond the limits of individual agents.

Theory & Application

History

The theory of computational agents goes back at least a quarter of a century when research in distributed artificial intelligence had been initiated. Agents made the real breakthrough in the nineties when the emphasis in mainstream research shifted: the focus on logic was extended and attention changed from goal-seeking to rational behavior; from ideal to resource-bound reasoning; from capturing expertise in narrow domains to re-usable and sharable knowledge repositories; from the single to multiple cognitive entities acting in communities (Russell and Norvig 1995). These developments also coincided with the evolution of network-based computing technology, the internet, mobile computing, the ubiquity of computing as well as novel, human-oriented software engineering methodologies (Luck et al. 2006). All these achievements led to what is considered now the agent paradigm of computing.

Characterization of agents

An agent operates in an **environment** from which it is clearly separated (see Figure 1). Hence, an agent makes observations about its environment, has its own knowledge and beliefs about its environment, has preferences regarding the states of the environment, and finally, initiates

Váncza, J.: Agent theory. *CIRP Encyclopedia of Production Engineering* (ed. Laperrière, L. Reinhart, G.), p. 10, http://www.springerreference.com/docs/html/chapterdbid/367479.html, Springer, (2013).

and executes actions to change the environment. Agents operate typically in environments that are only partially known, observable and predictable. **Autonomous** agents have the opportunity and ability to make decisions of their own. **Rational** agents act in the manner most appropriate for the situation at hand and do the best they can do for themselves. Hence, they maximize their expected utility given their own local goals and knowledge. Rationality can be bound by the computational complexity of a decision problem, the limitation of computing resources, or both. An agent with optimization objectives but with limited means is a **bounded rational** agent. A **reactive** agent responds in timely manner – in real or near-time – to changes of its environment, while a **proactive** agent is able to act in anticipation of future situations and goals.

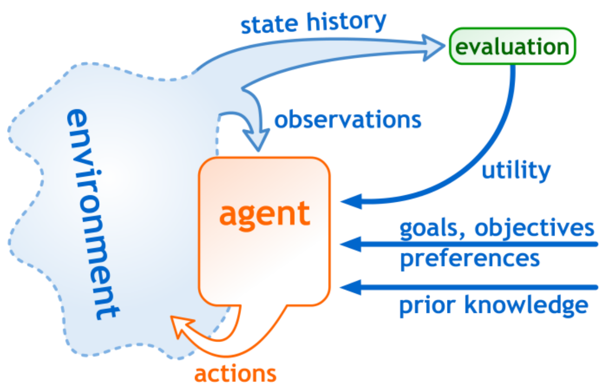

Figure 1: The agent and its environment (after Russell and Norvig 1995).

As for their key common properties, computational agents

- act on behalf of their designer or the user they represent in order to meet a particular **purpose**;
- are **autonomous** in the sense that they control both their internal state and behavior;
- exhibit some kind of **intelligence**, from applying fixed rules to reasoning, planning and learning capabilities;
- **interact** with their environment, and in a community, with other agents;
- are, ideally, **adaptive**, i.e., capable of tailoring their behavior to the changes of the environment without the intervention of their designer.

Further agent properties, characteristic in particular domains and applications, are **mobility** (when an agent can transport itself to another environment to access remote resources or to meet other agents), **genuineness** (when it does not falsify its identity), **transparency**, and credibility or **truthfulness** (when it does not communicate false information willfully). Even though they exhibit only some of the above properties, agents relax several strong assumptions of classical computational intelligence: they typically have incomplete and inconsistent knowledge as well as limited reasoning capabilities and resources.

Multi-agent systems

In a multi-agent system (MAS) the decisions and actions of various agents do necessarily interact. However, just due to **interaction**, a multi-agent system can occasionally solve problems that are beyond the limits of the competence of the individual agents and/or may exhibit

emergent behavior that cannot be derived from the internal mechanisms of the components. In a community an agent has to **coordinate** its actions with those of the other agents; i.e., to take the effects of other agents' actions into account when deciding what to do. Coordination models provide both **media** – such as channels, blackboards, pheromones, market, etc. – and **rules** for managing the interactions and dependencies of agents. Coordination requires some regulated flow of information between the agent and its surrounding environment, i.e., **communication**. In a MAS coordination is possible both by indirect communication via the environment, or by direct information exchange between specific agents. In any case, communication needs some **language(s)** with syntax and semantics, at least partially known for each communicating agent.

Collaboration means carrying out concerted activities so as to achieve some shared goal(s). For instance, in a scheduling domain machine agents may agree on executing each task of a job with the aim of completing an order by the given due date. The shared goal (completing an order) can be achieved only if all agents commit themselves to carrying out the actions they have agreed upon. In general, in a MAS of self-interested and autonomous agents meeting high-level objectives and satisfying system-wide constraints need **cooperation;** an interactive relationship that makes it possible to harness knowledge of other agents or to make use of their actions in the service of joint interests. The basis of any form of cooperation is reciprocity and trust between autonomous parties who can decide and act in their own right. Cooperation is the alignment of various, possibly even disparate goals in the hope of some mutual benefit. Cooperation can be developed among interrelated parties who have their own identity and discernible interests (expressed in terms of goals, objectives, utility or profit, etc.); who have the faculties for pursuing their own interest, and who admit to the autonomy of other, related parties. Cooperation has a number of forms in the physical and biological world, and is the prime basis of processes, organizations and institutions of human society (Axelrod 2006).

The overall operation of a multi-agent system is affected by an **organization** that is imposed on the individual agents. Even though there may be no global control or centralized data, and the computations are asynchronous, some organizational rules always exist. The organization determines the "sphere" of the activity of agents, as well as their potential interactions (see Figure 2).

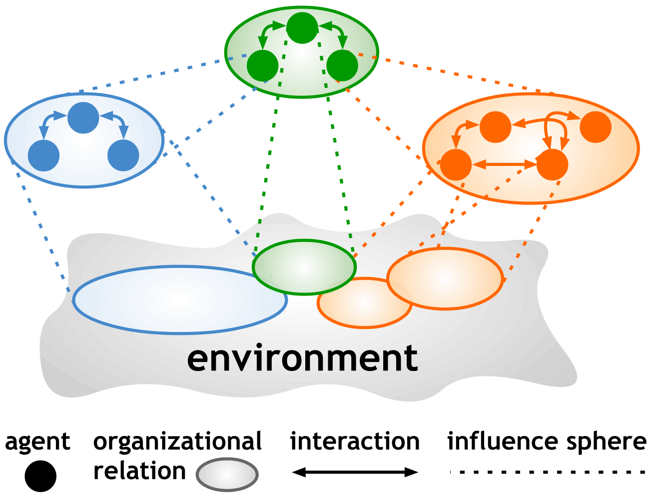

Figure 2: Generic scheme of a multi-agent system (after Jennings 2001).

The various organization patterns of multi-agent systems, such as teams, coalitions, markets, as well as hierarchic and heterarchic (including holonic) architectures provide different ways to achieve system-wide design objectives and/or to facilitate the emergence of desired properties in multi-agent systems.

Agent technologies

Agent level decision making

Depending on the actual problem and available knowledge at hand, agents can apply various **faculties of problem solving**, including searching, reasoning, planning, and learning. The notion of agents has a strong synthesizing power; hence the applied techniques may include both symbolic and sub-symbolic methods, classical and quantitative decision theory, as well as knowledge-based reasoning and sophisticated belief-desire-intention (BDI) model. There are several approaches for realizing agents:

- Following the principles of classical **decision theory**, an agent makes choices from a set of alternative actions so that it can maximize the expected utility of its decisions. If a utility function or the required input data are not available, qualitative models are applied that work with preferences.
- An agent may have explicit knowledge of how its actions can change the states of its environment. Given some states to achieve – so-called **goals** – the reasoning over future courses of actions is a key to intelligent behavior. Artificial intelligence provided a host of **planning** methods to solve this problem under various assumptions.
- The decision problem of an agent can be cast in terms of cognitive concepts, such as knowledge and belief, desires and goals, plans and intentions. A BDI agent is continually updating its beliefs based on perceptions, using its beliefs to reason about possible plans, committing to certain intentions, and realizing these intentions by acting. Agents who are situated in a dynamic environment can benefit from having **plans** which on the one hand, can constrain the amount of reasoning, and, on the other hand, can make coordination possible.

Interaction

Agents necessarily **interact** with each other – either indirectly, via the environment, or by direct communication. The various coordination and cooperation mechanisms range from emergent methods (without explicit communication among agents) to coordination protocols, coordination media and distributed planning (Luck et al. 2006). A **coordination media** provides a shared memory space for communicating data in an asynchronous way. Typical examples are blackboards, pheromones in stigmergy-based coordination, or attraction-repulsion field. **Coordination protocols** control the interactions of agents in order to reach common decisions. For instance, a widely used protocol is the **contract net protocol** where a manager agent makes arrangements via bidding to have some task performed by one or more other agents. Goal-oriented agents forming a community may have disparate and conflicting goals. For resolving conflict situations, various **negotiation** mechanisms were developed, including auctions, one-to-one negotiation, bargaining, and argumentation-based negotiation (Woolbridge 2001). Collaborative acting and planning involve the intentions of multiple agents. Since the presence of other agents is always a source of uncertainty (beyond other possible sources), collaboration requires an integrated treatment of the beliefs and intentions of the agents who may take part in a collaborative act. That is why the BDI model provided the theoretical basis for **agent communication languages** (ACL), including the widely used Foundation for Intelligent Physical Agents standard (see FIPA). Communication and interoperability requires consensual knowledge of a community. A so-called **ontology** is an explicit specification of the conceptual structures of a given domain. It is usually expressed in a logic-based language that makes it possible to distinguish classes, instances, properties, relations and functions in a clear-cut, consistent way. Consensual means that the whole community has a common understanding both on the content and form of the expressed knowledge. Ontologies also can facilitate machine processing: automated reasoning, as well as the inter-operability of different agents.

Organization

Like any community, a MAS is formed by agents that are aimed at achieving some purposes, be them individual, system-wide or both. It is no wonder that multi-agent systems adapt all the basic human organization patterns. There are a number of organization structures that define various patterns of decomposing work, assigning responsibilities to those who do the work, as well as collecting and combining results. A particular organization structure comes together with rules concerning the conditions of participating in a MAS, the assignment of the roles and relations, as well as the use of protocols - all of which together realize a particular **coordination mechanism**. In a MAS, however, human organizational models can be surpassed. The ideas of decentralized problem solving, including the resolution of the conflict between individual and collective good, are widely studied in a number of other disciplines, such as economics, game theory, political science, biology, and ecology. Computational models of agency borrowing analogies from these fields are similar in that they rely on some form of **self-organization**. No central control is exercised, and the systems adapt their structure and functionality to the changing requirements and environmental conditions. Typically, members of the systems are individually able to achieve simple tasks, but their interactions lead to the **emergence** of complex collective behavior (Ueda et al. 2001).

Use of agents

Agent technology and multi-agent systems, together with their supporting information and communication technologies – such as networking, software engineering, distributed and concurrent systems, mobile technology, electronic commerce, interfaces, semantic web, cloud computing – have found their way to application and deployment in many fields. The key success factors behind the use of agents are as follows (Luck et al. 2006, Monostori et al. 2006):

- Agents provide a new **design metaphor** for structuring knowledge (and system design, accordingly) around components that have autonomy and capability to communicate. Objects that earlier had complex properties can now be personified. Further to procedures, abstract data types, and objects, agents as intentional entities represent an increasingly powerful abstraction of computing. Any kind of intelligence requires the handling of **conflicts** rooted in disparate interests. Conflicts become explicit if the system is modeled like a community of self-interested agents. Hence, the agent-based approach forces the system's designer to find ways for managing conflicts.
- Agents and MAS provide a wide array of models, techniques, formal modeling approaches and development methodologies that all together shape general-purpose techniques of **agent-oriented software engineering** (AOSE) (Jennings 2001, Bordini et al. 2005). There have been developed several programming languages and software development environments which not only support MAS programming, but also implement key concepts of MAS in a unified framework.
- Agent-based modeling is especially suitable for **simulating** the behavior of complex systems operating in dynamic environments. In contrast to traditional, top-down approaches, the emphasis is on capturing the individuals, together with all their limitations and interactions. The question is whether and how local interactions can produce observable – and useful – patterns of global behavior. Hence, **agent-based simulation** (ABS) became an accepted methodology for developing plausible explanations for emergent phenomena. Alternatively, it was used for verifying multi-agent system design in a number of fields from engineering to natural and social sciences (Gilbert 2008).

Agent-based systems can be realized in a wide spectrum: from so-called coarse-grained agents (like BDI) with sophisticated communication, cognitive reasoning, and learning faculties up to fine-grained agents with a very limited operational repertoire, but high connectivity and intensive interactions. It is a general observation that fine-grained but complex systems may display patterns of behavior and develop certain functional properties that cannot be understood and explained solely on the basis of the control of the individuals. In the eye of the observer, these emergent features are unexpected, novel and show the traces of a stable order. Analysis of such systems leads to the terrain of **network theory**.

Agents in manufacturing

Enterprises always operated within the fabrics of economy, society and the ecosystem. In this complex, ever-changing, dynamic and hardly predictable environment manufacturing enterprises compete not only individually, but also as members of various networks. No wonder that manufacturing called for new, more robust, adaptable, fault-tolerant, decentralized and open organizational structures even before the paradigm of agent-based computing and MAS appeared (Hatvany 1985). Agent theories were really welcome because they helped to realize important properties as autonomy, responsiveness, redundancy, distributedness, and openness. Hence, many tasks related to manufacturing – from engineering design to supply chain management – have been to assigned to agents, small and large, simple and sophisticated, fine- and coarse-grained that were enabled and empowered to communicate and cooperate with each other (Monostori et al. 2006, Shen et al. 2006, Váncza et al. 2011).

Agents in engineering design and collaborative engineering

While design activities in various branches of engineering (mechanical, electrical, control, software, communication, etc.) are now being more and more integrated, **concurrent engineering** embraces all the main life-cycle activities such as marketing, product design, manufacturing, distribution, sales, operation, maintenance, disposal and re-cycling. Customers are involved in production from the decisive moment of the conception of ideas, already in the design of the product they are going to purchase. With the pervasive connectivity of the Internet, personalization has been increasingly adopted for consumer products.

In a MAS design problems and the available knowledge can be structured in appropriately distributed ways. Collaborative and concurrent approaches to design are successful in particular because they are utterly based on **interaction** that helps harness external knowledge that could not have been captured and internally represented. Furthermore, collaborative engineering makes explicit the disparate goals, objectives, priorities and concerns – in short, interests – of the various stakeholders related to a product's life-cycle. These interests manifest themselves as conflicts just in the early phase of design when decisions with far-reaching effects are made. Negotiation over conflicts can drive the design process towards innovative solutions (Lu et al. 2007). A trade of incomplete knowledge against interest can be very fruitful: rational, interest-seeking behavior on the part of autonomous of agents can result in successful overall performance even in cases when the agents have limited capabilities (knowledge and/or resources). At the same time, collaboration rests upon interaction, which is still the key to creative design.

Agents in production

Artifacts and related services are usually provided by **production networks** where autonomous enterprises are linked by relatively stable material, information and financial flows. The members that are cross-linked by information and communication technologies are not only able but also willing to interact with each other, i.e., exchange information about their products, intentions (plans), expectations (forecasts) and status. Agents offer adequate ways for modelling, controlling and simulating such complex production systems that should be **cooperative** and **responsive** (Váncza et al. 2011). As shown in Figure 3, the structure of such a network is defined by autonomous production nodes and logistics links (A). Each node has its own internal decision mechanism, typically on various levels of aggregation, from long-term sales and operations planning via medium-term production planning down to production scheduling and control (B). Finally, each node has its own execution mechanism where plans are realized on the shop floor (C). These three main levels—network, enterprise and shop floor—define a layered decision scheme where targets are set hierarchically, in a top-down way. On all levels, responsiveness requires timely decisions, though the timescales are consistent with the appropriate level. It is also essential to respond both to new or altered demands (coming usually from an upper level) and changes and disruptions (feedback from a lower level). At all levels, the system has to be robust in face of various disturbances coming from the environment.

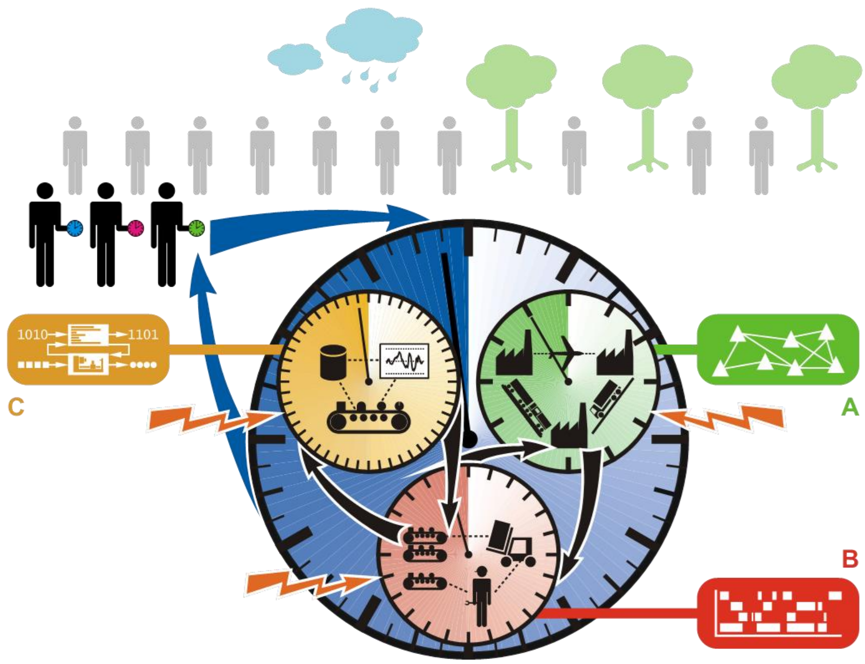

Figure 3. Multi-level enterprise model (after Váncza et al., 2011).

Industrial applications

There are some barriers for the industrial take-up of agent technologies, such as scalability, inherent complexity, safety, risk of inconsistent global operation, the appearance of inevitable conflicts between self-interested entities, and the extra burden of communication. Until recently, the industrial acceptance of MAS in manufacturing has not kept up with expectations, partly because of the above issues, and partly because of the difficulties in their stepwise integration with existing legacy systems. Against all the above difficulties, there is a consensus concerning the application areas of highest potential:

- Where neither access to information nor decision rights can be centralized. This is the case in managing supply networks, including transportation and material handling.
- In complex operations management – such as resource allocation, planning and scheduling – where the problems can be decomposed along distinct goals and performance objectives.
- In industrial monitoring and control where robustness and fast reconfigurability are essential requirements in a distributed setting.

All in all, agents hold the promise for resolving compelling challenges that are rooted in generic conflicts between competition and cooperation, local autonomy and global behavior, design and emergence, planning and reactivity, uncertainty and a plethora of information (Váncza et al.

2011). Developments in various agent technologies are still extremely dynamic, innovative and ramifying. At the same time, there is also a strong commitment to convergence with current industrial software technologies. The evolution of multi-agent systems and manufacturing will proceed hand in hand: the former can receive real challenges from the latter, which, in turn, will have more and more benefits in applying agent technologies.

Cross References

Adaptability, Artificial Intelligence, Autonomous Production Control, Decentralization, Distributed Manufacturing, Emergence, Holonic Manufacturing Systems, Open Architecture

References

Axelrod R (2006) The evolution of cooperation. (revised ed.), Perseus Books Group

Bordini RH, Dastani M, Dix J, El Fallah Seghrouchni A (eds.) (2005) Multi-agent programming: Languages, platforms and applications. Springer

FIPA: Foundation for Intelligent Physical Agents, www.fipa.org

Gilbert N (2008) Agent-based models. SAGE Publications

Hatvany J (1985) Intelligence and cooperation in heterarchic manufacturing systems. *Robotics and Computer-Integrated Manufacturing* 2(2):101–104

Jennings NR (2001) An agent-based approach for building complex software systems. *Communications of the ACM* 44(4):35–41

Lu SY-C, Elmaraghy W, Schuh G, Wilhelm R (2007) A scientific foundation of collaborative engineering. *CIRP Annals - Manufacturing Technology* 56(2):605–634

Luck M, McBurney P, Shehory O (2005) Agent technology: Computing as interaction. A roadmap for agent-based computing. AgentLink III, http://www.agentlink.org/roadmap/al3rm.pdf

Monostori L, Váncza J, Kumara SRT (2006) Agent-based systems for manufacturing. *CIRP Annals - Manufacturing Technology* 55(2):697–720

Russel S, Norvig P (1995) Artificial intelligence: A modern approach. Prentice Hall, third edition (2010)

Shen W, Hao Q, Yoon H-J, Norrie DH (2006) Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced Engineering Informatics* 20(4):415–431

Shoham Y, Leyton-Brown K (2009) Multiagent systems: Algorithmic, game-theoretic, and logical foundations. Cambridge University Press, Cambridge, UK

Ueda K, Márkus A, Monostori L, Kals HJJ, Arai T (2001) Emergent synthesis methodologies for manufacturing. *CIRP Annals - Manufacturing Technology* 50(2):535–551.

Váncza J, Monostori L, Lutters D, Kumara SRT, Tseng M, Valckenaers P, Van Brussel H (2011) Cooperative and responsive manufacturing enterprises. *CIRP Annals - Manufacturing Technology* 60(2):797–820

Wooldridge P (2000) An Introduction to multiagent systems. Addison-Wesley, Reading, MA