Current Development in Artificial Intelligence.
ISSN 0976-5832 Volume 6, Number 1 (2015), pp. 1-7

http://www.irphouse.com

Software Agent Technology

Anjali Kedawat

Lecturer, Amity University Rajasthan
E-Mail: anj_ked@yahoo.co.in

Abstract

Software Agent Technology is fastly growing as a powerful Technology. Software Agent Technology is for the development of Complex Software systems, synthesizing contributions from the many different research areas including the software engineering, artificial intelligence, distributed computing and the robotics. Developing applications in the terms of the autonomous software agents that will exhibit the intelligent behaviors and the proactive, and also they interact with one another in the terms of high-level protocols and the languages leads to the new programming paradigm. Agent based Software Engineering offers the same promising benefits over used in the mainstream software Engineering. First up all, it promises to solve the interaction bottle neck decreasing complexity of the Engineering Software. Secondly it promises to decrease the difficulties involved in the designing systems for the complex environments. The later claim is based on Qualitative Argumentation that multi agent systems are well suited for that environments. Or at least that agent oriented mindset promotes the good design practice.

Keywords: autonoumous, BDI, reactive, deliberative

1. INTRODUCTION

Agent-based computing has been hailed as the next significant breakthrough in software development, with the potential to affect many aspects of computer science, from artificial intelligence to the technologies and models of distributed computation. Agent-based systems are capable of autonomous action in open, dynamically-changing environments. Agents are currently being applied in domains as diverse as business information systems, computer games and interactive cinema, information retrieval and filtering, user interface design and industrial process control. The aim of the Agent based computing is to bring together researchers and developers from industry and academics in order to report on the latest scientific and technical advances, discuss and debate the major issues, and showcase the latest systems

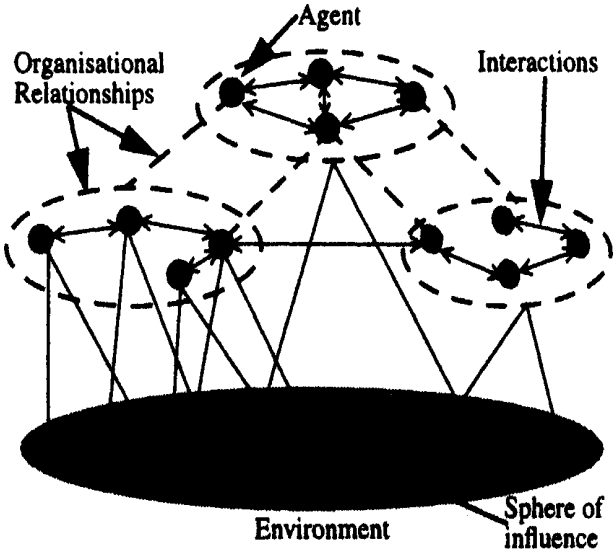

Fig.1 Canonical Agent Based System

2 KEY PROPERTIES of an AUTONOUMOUS 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 quickly 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.
- 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.

3 AGENT CONCEPTS

3.*1* *Agents use models:*

- agents as design metaphor, 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.

3.2 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).

4. SIMILAR CONCEPTS

41. Objects :

computational entity with its encapsulated state, ability toper form 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

42. Expert system[2]–

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

5. REASEARCH CHALLENGES and APPLICATIONS of AGENT BASED COMPUTING

- 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.
- 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

- 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.
- 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.

6. AGENT APPLICATIONS

6.1 Distributed and collective aspects[1] :

- 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-real-time) manufacturing or industrial systems control, with re-planning, or fast local reconfiguration)

6.2 Simulation and modeling scenarios[1]:

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.

7. PROBLEM 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 teleportation with autonomous decision making.
- utility networks[3]: Energy distribution networks, mobile operators networks, cable provider networks - simulation and predication of alarm situations, prevention to black-out and overload, intrusion detection.

8. AGENT ARCHITECTURES

- **Reactive agents*[4]*** are agents that contain no symbolic knowledge representation (i.e: no state, no representation of the environment, no representation of the other agents,...). Their behavior is defined by a set of perception-action rules.

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

- **Deliberative agents[5]** 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 Belief 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.

9. FORMAL MODEL of AGENCY*[6]*

Agent is an encapsulated computational system, that is situated in some environment, and that is capable of flexible, autonomous behavior 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

10 CONCLUSION

It is probably fair to say that (multi-)agent technology is one of the most hyped software technology of the late 1990s. While the technology has much to offer, it is important not to oversell it.. Multi-agent technology is likely to be most useful for a specific class of applications (albeit an important class).(Thus agents are *not* likely to replace objects!) One of the most worrying aspects of the current interest in agents is to label *everything* as an agent: one finds academic articles describing theorem provers as 'theorem proving agents', planners as 'plan development agents', and even numerical algorithms as 'numerical algorithm agents'. Many people strongly believe that the notion of an agent as an independent rational decision maker is of great value not just in AI, but also in mainstream computer science. Similarly, the techniques being developed in multi-agent systems research, to enable agents to cooperate and negotiate, are surely of fundamental importance for the future.

References

1. Sixth National Conference on Artificial Intelligence (AAAI-87), pages 268–272, Seattle, WA, 1987.

2. A. H. Bond and L. Gasser, editors. Readings in Distributed Artificial Intelligence. Morgan Kaufmann Publishers: San Mateo, CA, 1988.

3. [Wegner, 1997] P. Wegner. Why interaction is more powerful than algorithms. *Comms. of* ACM 40(5):8O-91, 1997.

4. Wooldridge, 1997] M. Wooldridge. Agent-based software engineering. *IEE Proc Software Engineering* 144:26-37, 1997.

5. [Wooldridge and Jennings, 1995] M. Wooldridge and N. R. Jennings. Intelligent agents: theory and practice. The *Knowledge Engineering Review* 10(2): 115-152, 1995.

6. [Wooldridge and Jennings, 1998] M. J. Wooldridge and N. R. Jennings. Pitfalls of Agent-Oriented Development. *Proc 2nd Int. Conf on Autonomous Agents,* Minneapolis, USA, 385-391, 1998.