CPSC 589

Seminar in Computer Science

Agent-based Software Engineering

Chang-Hyun Jo
Department of Computer Science
California State University Fullerton
jo@ecs.fullerton.edu

Contents

We are here

Agent-based Computing

- Agent-based computing is involved in development of robust and scalable software systems that requires autonomous agents that can complete their objectives while situated in a dynamic and uncertain environment, that can engage in rich, high-level social interactions, and that can operate within flexible organizational structures [Jennings 2000].

Agent-based Computing

- Agent-based computing opens a new genre by synthesizing Computer Science and especially Artificial Intelligence [Jennings 2000].
- Agent-based computing has the potential to improve both the theory and the practice of modeling, designing, and implementing computer systems [Jennings 2000].

Agents

- An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives [Jennings 2000].

Agents

- Clearly identifiable problem solving entities with well-defined boundaries and interfaces
- Situated (embedded) in a particular environment – they receive inputs related to the state of their environment through sensors and they act on the environment through effectors
- Designed to fulfill a specific purpose – they have particular objectives (goals) to achieve

Agents

- Autonomous – they have control both over their internal state and over their own behavior
- Capable of exhibiting flexible problem solving behavior in pursuit of their design objectives – they need to be both reactive (able to respond in a timely fashion to changes that occur in their environment) and proactive (able to act in anticipation of future goals)

Multi-agents

- Agents mostly need to interact with one another, either to achieve their individual objectives or to manage the dependencies that ensue from being situated in a common environment [Jennings 2000].

Multi-agents

- Research issues [Jennings 2000]:
  - Devise protocols that enable organizational groupings to be formed and disbanded
  - Specify mechanisms to ensure grouping act together in a coherent fashion
  - Develop structures to characterize the macro behavior of collectives

Techniques for Tackling Complexity in Software

- Decomposition
  - Divide problems into smaller, more manageable chunks, each of which can be dealt with in relative isolation
- Abstraction
  - Define a simplified model of the system that emphasizes some of the details or properties, while suppressing others
- Organization (Hierarchy)
  - Identify and manage the interrelationships between the various problem solving components
- *Booch, G. Object-Oriented Analysis and Design with Applications, Addison-Wesley, 1994.*

Decomposition

- The agent-oriented approach advocates decomposing problems in terms of autonomous agents that can engage in flexible, high-level interactions.
- Considering the autonomous nature of the agents first.
  - The problem solving entities have their own persistent thread of control.
  - Autonomous entities decide for themselves which actions they should perform at what time.
    - All real systems are distributed. (multiple threads)
    - Real systems have no top. (decentralization)

Abstraction

- A significant part of any design process is finding the right models for viewing the problem.
  - In general there will be multiple candidates and the difficult task is picking the most appropriate one.
  - The good abstractions are those that minimize the semantic distance between the units of analysis and the constructs.

Abstraction

- The problem to be characterized consists of subsystems, subsystem components, interactions and organizational relationships.
  - Strong degree of correspondence between the notions of subsystems and agent organizations
  - The interaction between the subsystems is naturally viewed in terms of high level of social interactions. (cooperating to achieve common goals, coordinating their actions, negotiating to resolve conflicts, …)
  - Organizational relationships (roles, norms, social laws) (modeling collectives – joint intentions, teams)

Organization

- Organizational constructs are first-class entities in agent systems.
  - Explicit representations of organizational relationships and structures
  - Computational mechanisms for flexibly forming, maintaining and disbanding organizations

Agent-Oriented Software Engineering

- Will AOSE succeed as a main stream SE paradigm? [Jennings 2000]
  - The agent-based approach can be viewed as a natural next step in the evolution of a whole range of approaches to SE.
    - Providing a higher level of computational abstraction
    - Realized through object-based systems or in a component-based fashion
  - Agent-based techniques are the ideal computational model for developing software for open, networked systems.

Agent-Oriented Software Engineering

- Agent computing views the world as a set of autonomous agents that collaborate to perform some higher level function.
- The basic build blocks of the programming models exhibit increasing degrees of localization and encapsulation.
  - Agents localize purpose inside each agent, give each agent its own thread of control, and encapsulate action selection.

Agent-Oriented Software Engineering

- Agents enable whole subsystems and flexible interactions to be reused.
  - Design patterns
  - Application frameworks
  - Agent designs and implementations are reused within and between applications.

Agent-Oriented Software Engineering

- BDI architecture
  - Belief
    - What the agent knows
  - Desire
    - What the agent wants
  - Intention
    - What the agent is doing
- BDI Examples
  - Air traffic control, process control, simulation, fault diagnosis, transportation, scientific data interpretation, …

Agent-Oriented Software Engineering

- Entities in open, networked systems
  - Active and autonomous
  - Reactive and proactive
  - Engaging in flexible interactions
  - Organizational relationships

Downside of an Agent-based Approach in SE

- General aspects and difficulties of complex system development
  - Difficulties arisen from engineering large systems per se
    - Performance engineering and security
  - Distributed and concurrent systems
  - Maintaining an ongoing interaction with a dynamic and unpredictable environment

Downside of an Agent-based Approach in SE

- Issues intrinsic to the agent-based philosophy
  - Social and pragmatic problems associated with developing systems using a new technology
- Two major drawbacks
  - The patterns and the outcomes of the interactions are inherently unpredictable.
  - Predicting the behavior of the overall system based on its constituent components is extremely difficult because of the strong possibility of emergent behavior.

Downside of an Agent-based Approach in SE

- Source of unpredictability
  - The nature (a simple request versus a protracted negotiation) and the outcome of an interaction cannot be determined at the onset.
  - Unpredictability in agent-oriented systems related to the notion of emergent behavior.

A Social Level View

- A social level characterization of agent-based systems proposed by Jennings [2000]
  - Comparing with Newell's Knowledge Level Analysis
  - Enabling the overall system's behavior and key conceptual structures to be studied without the need to delve into the implementation details of the individual agents or the specifics of particular interaction protocols.
  - Thus prediction of the behavior of the social agents and of the overall system can be made more easily.

A Social Level View

Table 1
Summary of the knowledge and social levels

Dimension	Description	Knowledge level	Social level
System	Entity to be described	(asocial) Agent	Agent organisation
Components	The system's primitive elements	Goals, Actions	Agents, Interaction channels, Dependencies, Organisational relationships
Compositional law	How the components are assembled	Various	Roles, Organisation's rules
Behaviour law	How the system's behaviour depends upon its composition & components	Principle of rationality	Principle of organisational rationality
Medium	The elements to be processed to obtain the desired behaviour	Knowledge	Organisation and social obligations, Means of influencing others, Means of changing organisational structures

- Characterization follows Newell's basic nomenclature for specifying computer system levels.

References

- Acknowledgement
  - These notes are summarized mainly from the following references.
  - Jennings, Nicholas R., On agent-based software engineering, *Artificial Intelligence,* volume 117, 277-296, February 2000.

References

- Agent Based Engineering (ABE) Group, Center for Design Research, Stanford University, http://www-cdr.stanford.edu/ABE/, 2003.
- Bratman, M. E., Intention, Plans, and Practical Reason, *Harvard University Press*, 1987. (also available from CSLI Publications, 1999)
- DeLoach, Scott A. Multiagent Systems Engineering: A Methodology and Language for Designing Agent Systems, http://en.afit.af.mil/ai/publications/Conference/aois-99/MaSE-AOIS99.htm, 1999.
- Depke, Ralph, Heckel, Reiko, and Kuster, Jochen, Improving the Agent-Oriented Modeling Process by Roles, *AGENTS'01*, 640-647, June 2001.
- Einhorn, M. Jeffery, A BDI Agent Software Development Process, MS Thesis, (Advisor: Chang-Hyun Jo), Department of Computer Science, University of North Dakota, May 2002.
- Elammari, M. and Lalonde, W. An Agent-Oriented Methodology: High-Level and Intermediate Models, Proc. of the Agent-Oriented Information Systems (AOIS) 1999, Heidelberg, Germany, June 1999, Also available at http://citeseer.nj.nec.com/cache/papers/cs/15637/http:zSzzSzwww.carleton.cazSz~elamarizSzAOIS99zSzaom-aois99.pdf/elammari99agentoriented.pdf, 1999.
- Foner, Lenny. What's an Agent, Anyway? A Sociological Case Study, http://lcs.www.media.mit.edu/people/foner/Julia/Julia.html, 1999.

References

- Franklin, Stan and Graesser, Art. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, http://www.msci.memphis.edu/~franklin/AgentProg.html, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.
- Gamma, E., r. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object Oriented Software, *Addison-Wesley*, 1995.
- Hayden, Sandra, Carrick, Christina, and Yang, Qiang, A Catalog of Agent Coordination Patterns, ACM Press, 412-413, 1999.
- *A Special Edition for Agents, IEEE Internet Computing* Vol 1., No. 4, July/August 1997.
- Iglesias C. A., Garijo M, Gonzalez J. C., and Juan R. Velasco, Analysis and Design of Multiagent Systems using MAS-CommonKADS, In M.P. Singh, A. Rao, and M.J. Wooldridge, editors, *Proc. 4th Int. Workshop on Agent Theories, Architectures, and Languages (ATAL-97)*, volume 1365 of LNAI, 313-328, Springer-Verlag, July 24-26, 1998.

References

- Janson, S. Agent-based Systems, http://www.agentbase.com/survey.html, 2003.
- Jennings, Nicholas R., On agent-based software engineering, *Artificial Intelligence,* volume 117, 277-296, February 2000.
- Jennings, Nicholas R., An Agent-Based Approach for Building Complex Software Systems, *Communications of the ACM*, 44(4), 35-41, April 2001.
- Jo, Chang-Hyun. A Seamless Approach to the Agent Development, ACM 2001 15th Annual Symposium on Applied Computing (ACM SAC'01), Las Vegas, 641-647, (March 2001).
- Jo, Chang-Hyun and Allen J. Arnold, "Agent-based Programming Language: APL", ACM 2002 16th Annual Symposium on Applied Computing (ACM SAC'02), Madrid, Spain, 27-31, (March 2002).
- Jo, Chang-Hyun. "A New Way of Discovery of Belief, Desire and Intention in the BDI Agent-Based Software Modeling", The International Conference and Exhibition on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM 2003), Manila, Pavillon Hotel, March 27-30, 2003. (ISBN# 971-92723-0-9)
- Jo, Chang-Hyun and Einhorn, Jeffery M., "A Process for BDI Agent-Based Software Construction", The 2003 International Multi-Conference in Computer Science and Computer Engineering – The International Conference on Software Engineering, (IMCCSCE – SERP'03), 204-209, Las Vegas, Nevada, June 23-26, 2003.

References

- Jo, Chang-Hyun, Zhao, Wei and Cong, Bin., "A Design and Implementation of the Belief-Desire-Intention Agent-based Programming Language", ***Information: An International Interdisciplinary Journal***, ISSN 1343-4500 (print), 1344-8994 (electronic), 7(1), International Information Institute, (January 2004), 137-153.
- Jo, Chang-Hyun. "A New Way of Discovery of Belief, Desire and Intention in the BDI Agent-Based Software Modeling", ***the International Journal of Advanced Computational Intelligence & Intelligent Informatics (JACIII)***, ISSN 1343-0130, 8(1), 2-6, Jan. 2004.
- Jo, Chang-Hyun, Guobin Chen and James Choi, "A New Approach to the BDI Agent-Based Modeling", ACM SAC 2004, 1541-1545, Nicosia, Cyprus, March 14-17, 2004.
- Jo, Chang-Hyun and Einhorn, Jeffery M., "A BDI Agent-Based Software Process", ***Journal of Object Technology (JOT)***, Vol.4, No.9, 101-121, November – December 2005. (available at http://www.jot.fm/issues/issue_2005_11/article3)
- Jo, Chang-Hyun, Research related to Agent Computing, http://jo.ecs.fullerton.edu/research, 2006.
- Kinny, D., Georgeff, M., and Rao, A. A Methodology and Modelling Techniques for Systems of BDI agents. In Agents Breaking Away: Proc. Of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, (LNAI Vol. 1038): 56-71, Springer-Verlag, 1996, also at http://citeseer.nj.nec.com/cache/papers/cs/40/ftp:zSzzSzwww.aaii.com.auzSzpubzSzaaii-technoteszSztechnote58.pdf/kinny96methodology.pdf.

References

- Maes, Pattie. Autonomous Agents Group, MIT Media Laboratory, http://agents.media.mit.edu/index.html, 2003.
- Maudlin Michael. Julia, Information available from Forner, Lenny's Julia, http://foner.www.media.mit.edu/people/foner/Julia/, 1999.
- Petrie, Charles J. Agent-based Engineering, the Web, and Intelligence, http://www-cdr.stanford.edu/NextLink/Expert.html, 2003, also available from IEEE Expert, Dec. 1996.
- Petrie, Charles, Agent-Based Software Engineering, *Agent-Oriented Software Engineering, Lecture Notes in AI, Springer-Verlag,* 58-76, 2001.
- Rao, Anand S. and Georgeff, Michael P., BDI Agents: From Theory to Practice, *Australian Artificial Intelligence Institute*, April, 1995.
- Sycara, K. P. Multiagent Systems, http://www-2.cs.cmu.edu/~softagents/papers/multiagentsystems.PDF, also appears in AI Magazine, 19(2), 79-92, 1998.
- UMBC Agent Web, News and Information on software agent technology, http://agents.umbc.edu/, 2003.
- Weiss G., editor, Multi-Agent Systems, *The MIT Press: Cambridge, MA*, 1999.

References

- Wooldridge, Michael and Nicholas R. Jennings, "Agent Theories, Architectures, and Languages: a Survey," in Wooldridge and Jennings Eds., Intelligent Agents, Berlin: Springer-Verlag, 1-22. 1995.
- Wooldridge, M. and Jennings, N. R., Intelligent Agents: Theory and Practice, *Knowledge Engineering Review, Cambridge Univ. Press,* 10(2), 115-152, June 1995.
- Wooldridge, M. and Jennings, N. R., Software Engineering With Agents: Pitfalls and Pratfalls, *IEEE Internet Computing*, 20-27, May-June 1999.
- Wooldridge, M., Jennings, N. R., and Kinny, D., A Methodology for Agent-Oriented Analysis and Design, *Autonomous Agents 1999, Seattle, WA,* 69-76, 1999.
- Wooldridge, M., Reasoning about Rational Agents, *The MIT Press: Cambridge, MA*, 2000.