B.H. Far's Research
© B.H. Far
far@ucalgary.ca
http://www.enel.ucalgary.ca/People/far

Theme 3: Methodological support for interactive software agents

Abstract

Concept: Software agents are heterogeneous (i.e., developed by various vendors and based on different architecture and technology), scalable and networked software systems for modern complex tasks. When building software agents we usually design friendly programs that fit into the design perfectly, divide the job among themselves, work cooperatively in order to accomplish a task, in a similar way that distributed/parallel processing does. However, such software systems may fail to deliver the required functionality in a society in which tasks are complex and resources are limited. What if the assumption of being cooperative and coordinative no longer applies? What if the decision points in the algorithms become probabilistic rather than deterministic? What if the interactions between programs are neither static (e.g., using stubs, remote method invocation RMI, etc.) nor dynamic (e.g., using naming services, yellow pages, etc.) but opportunistic? These are the driving points of this research. In this project we focus on quality and reliability of software agents that are neither cooperative nor coordinative, but competitive. The agents may compete for limited resources and decide upon the next operation to execute based on probabilistic sensory data, taking the factors of risk and chance. Probabilistic interaction, decision making based on uncertainty and risk, and collective reliability are three main factors that are missing from contemporary software engineering paradigms.

Goal: The goal of the proposal is to devise theories, techniques and measures for enhancing quality and reliability of agent-based systems.

Uniqueness: (a) Focus on agent system reliability and quality. (b) Focus on agent interactions by starting with a complete set of possible interactions among software agents. (c) Focus on decision making based on multiple threads of control rather than reasoning based on a single thread of control.

1. Introduction

The software marketplace recognizes that successful software products need continual change in order to respond to competitive constraints as well as to address requirements changes, exploit new technologies, and accommodate changing user expectations. Traditional views of software products and conventional methods of software development are gradually replaced in the commercial community by more flexible and robust ones. Agent technology and agent-based software development have been seen as alternatives.

Interaction-based decision making is a central issue in multi-agent systems (MAS). Software agents interact with each other to solve problems; to share expertise; to work in parallel or sequence on common problems; to represent multiple viewpoints and the knowledge of multiple experts; and to compete for limited resources. Currently, many of those interaction-based decisions are made using pre-defined algorithms or static knowledge bases, based on simplistic rules of thumb, and with limited links to best knowledge or models. This has a negative impact on the overall reliability and quality of service (QoS) of the MAS, which is a central concern in the commercial community.

The main characteristics of *interaction-based decision-making* in MAS are: (C1) the quantity and quality of information available is typically low; (C2) the observation cost of collecting information is high; (C3) the processes and available decision parameters are dynamically changing; (C4) the resources are limited and decisions are made in a competitive environment.

This project views the *interaction-based decision making* in MAS from a software engineering perspective. We particularly focus on quality and reliability of software agents that are either cooperative and coordinative, or competitive. The agents may compete for limited resources and decide upon the next operation to execute based on probabilistic sensory data, taking the factor of risk. *Probabilistic interaction*, *decision making based on uncertainty and risk*, and *collective reliability* are three main issues that are missing from contemporary software engineering paradigms. Such issues are studied separately in artificial intelligence (AI), operational research (OR) and software engineering (SE) disciplines.

This proposal enhances the previous projects and builds upon their results by: (1) Focusing on quality and reliability issues of multi-agent systems. (2) Devising a complete set of agent interaction scenarios. This is a superset of the conventional 3 scenarios (cooperation, coordination and competition) of the previous projects. (3) Identifying the decision making mechanisms for each scenario. The mechanisms are not only game-theoretic (as in the previous projects) but also based on other methods such as maximum entropy, Bayesian networks, etc. (4) Building a library of decision making modules. The library will be extended and enhanced to account for all the scenarios and new mechanisms.

2. Objectives

Long-term (ultimate) objectives:

- To maintain the Canadian leading role of research in MAS and its applications in the ICT industries.
- To train HQP and transfer agent-based technology and agent-based system development methodologies to the ICT industries.

Short-term objectives:

The short-term goal of this proposal is to devise theories, techniques and measures for enhancing quality and reliability of agent-based systems. This project is a step towards this goal. This can be broken down into four sub-goals:

1) ***To formalize collective reliability for multi-agent systems:*** Software system reliability is usually defined as the probability that a system functions without failure for a specified time in a specified environment [MUSA1998]. Unfortunately this definition lacks the notion of purpose or goal, which is essential to agent-based systems. Furthermore, the overall reliability of a MAS relies on the reliabilities of its constituent agents and trade-offs among them, i.e., *MAS collective reliability*. The traditional reliability engineering models and techniques must be enhanced to account for these two issues. Therefore this sub-goal can be specified as:
    - **G11:** Defining and devising techniques for specifying and measuring goal-oriented reliability.
    - **G12:** Devising models and techniques for specifying and measuring MAS collective reliability.

2) ***To Devise models for interaction-based decision making in multi-agent systems:*** The goal is to devise techniques to handle the (C1) low quality initial information; (C2) reduce the observation costs; and (C3) handle dynamically changing decision parameters in (C4) competitive environments. Therefore this sub-goal can be specified as:
    - **G21:** Devising agent model and techniques for interaction-based decision making based on *complete knowledge assumption*, that is, the other agents' goals, strategies (i.e., actions to select from) and utilities (i.e., the pay-offs of actions) are known.
    - **G22:** Devising agent model and techniques for interaction-based decision making based on incomplete knowledge and risk assumptions, that is, the other agents' goals, strategies and utilities are unknown but their probability distribution are known.
    - **G23:** Devising agent model and techniques for interaction-based decision making based on incomplete knowledge and degree of optimism assumptions, that is, the other agents' goals, strategies and utilities are unknown and their probability distribution are also unknown.

3) ***To develop a catalogue of agent decision making mechanisms and supports techniques for interaction scenarios:*** That is to map each interaction scenario (45 identified scenarios) to a number of decision making techniques appropriate for that specific scenario and specify the scenarios and the decision making techniques.
    - **G31:** Developing the MAS decision making catalog and its specification.

4) ***To Develop and enhance a library of reusable agent decision making modules:*** That is to revise the design and implement a commercial-off-the shelf (COTS) library of reusable software components that can be used to build reliable multi-agent systems for real business applications.
    - **G41:** Implementing the MAS decision making commercial-off-the shelf (COTS) library.

3. Literature Review

This project is the intersection point of software engineering (SE), reliability engineering, decision theory, and multi-agent system research. Research in the following areas is particularly important to this work:

3.1 Multi-agent system (MAS) research

Traditional software systems can handle data and information. Data is defined as a sequence of quantified or quantifiable symbols. Information is about taking data and putting it into a meaningful pattern. Knowledge is the ability to use that information. Knowledgeability includes representing knowledge about the external world, reasoning with it and sharing it. Various techniques and methodologies to handle the knowledgeability, autonomy and interactivity have already been introduced [JENNINGS2000], [WEISS2000], [SUBRAMAN2000].

Agent-based systems are different from conventional software systems in eight basic points [WEISS2000], [FAR2003]: (1) When designing agent one defines "roles" (i.e., what to do) for them but in software one assigns "tasks" (i.e., what and how to do). (2) Emergent (unpredicted run-time) behavior may be present in MAS. (3) In MAS the knowledge cycle starts with observation and data collection while in software the data is usually given. (4) In MAS the interactions among constituent agents are comparable in size to interactions within an agent. (5) In MAS symbol level communication and common ontology is needed to share expertise or to interact intelligently. (6) Software components must have complete information of each others existence and services (i.e., knowledge completeness assumption). This may not be true in MAS. (7) In software systems, decision points in algorithms are deterministic. In MAS, decision point may be probabilistic. (8) In MAS each constituent agent can initiate a decision making process and the MAS overall decision making is based on multiple threads of control.

Four main approaches for multi-agent system design are reported: The MAS-common KADS [IGLESAIS1997]; the MASSIVE development method [LIND2000]; the agent-oriented analysis and design method [WOOLDRIDGE1998]; and development based on the Beliefs-Desires-Intentions (BDI) [RAO1995]. The final two are commonly cited and quite popular. In [WOOLDRIDGE1998] a conceptual framework - composed of agent roles, permissions, responsibilities and protocols - is proposed that maps to a methodological model for analysis (interactions and roles model) and design (agent, service and acquaintance models). Georgeff and Rao claim that the widely referenced *Belief-Desire-Intention* (BDI) notation is necessary in design of complex and real-time multi-agent systems [RAO1995] and many research papers have been emerged from this idea. There are recent efforts in the Modeling Technical Committee (M-TC) of Foundation for Intelligent Physical Agents (FIPA) to put the agent-based systems into the context of object-oriented system development and enhance the unified modeling language (UML) to account for agent specific concepts (AUML) [ODELL2001].

3.2 Decision Theory

John M. Keynes (1883-1946) has defined *risk* as a situation in which the probabilities of the various possible states are known; and *uncertainty* as a situation in which probabilities of the various possible states are unknown. Decision making under risk is studied extensively in risk analysis and economy [CHIANG84]. There are a number of decision making techniques based on uncertain data, information and knowledge. A modest list may include: uncertainty management using certainty factors [SHORTLIFFE1979]; Bayesian belief networks (BBN) and dynamic BBN [PEARL1990]; game theory, including static games of complete information; static games of incomplete information (static Bayesian game); dynamic games of complete and perfect/imperfect information; dynamic games of incomplete information [HARSANYI1968, OSBORNE1994, KUHN1997, BIERMANN1998]; decision making models under uncertainty and risk including dominance, maximax, maximin and minimizing the regret [FRENCH1986]; Dempster-Shafer's Theory of Evidence [SHAFER1976]; Ordered Weighted Averaging (OWA) [YAGER1988]; and uncertain programming techniques (linear programming, nonlinear programming, multi-objective programming, goal programming, integer programming, multilevel programming, dynamic programming, expected value models, chance-constrained programming, dependant-chance programming) [LIU2002]. Each of the techniques is appropriate for a certain class of agents' interaction scenarios.

3.3 Collective reliability for multi-agent system

Software reliability and testing has been studied from both computer scientific and engineering point of view. Challenges include efficient modeling software reliability [LYU1986] and engineering reliability process [MUSA1998] and tailoring the process to modern software development, such as object-oriented, incremental, iterative, agile, agent-based and component-based development [KAN2002]. Balance among the time-budget-quality trio [MUSA1998] was convincing enough for the past two decades. Nowadays two more and equally important attributes, namely, *people* and *technology*, are added and incorporating reliability measures related to them are equally important.

4. Methods and Proposed Approach

Phase 1 implementing Objective G11 and G12: In conventional software reliability theory, reliability may increase only if a fault is found and removed and the necessary reliability is defined with respect to failure intensity objective for the product. In order to implement **G11** (defining and devising techniques for specifying and measuring goal-oriented reliability) we will enhance the common reliability models and measures to give different severity weights to the faults that are correlated with the agent's goal. As for **G12** (devising models and techniques for specifying and measuring MAS collective reliability) we revise the definition of operational profile and the reliability engineering process to account for collective reliability. The technique can also be used for reliability of component-based software systems.

Phase 2 implementing Objective G21, G22 and G23:
For **G21**, **G22** and **G23** we must devise an agent model that accounts for the probabilistic knowledge on the values of attributes (goals, strategies, utilities, etc.) of the other agents. A popular model for software agents is to model them as software components [PARUNAK2002], i.e., a software package with complete encapsulation of its behavior that

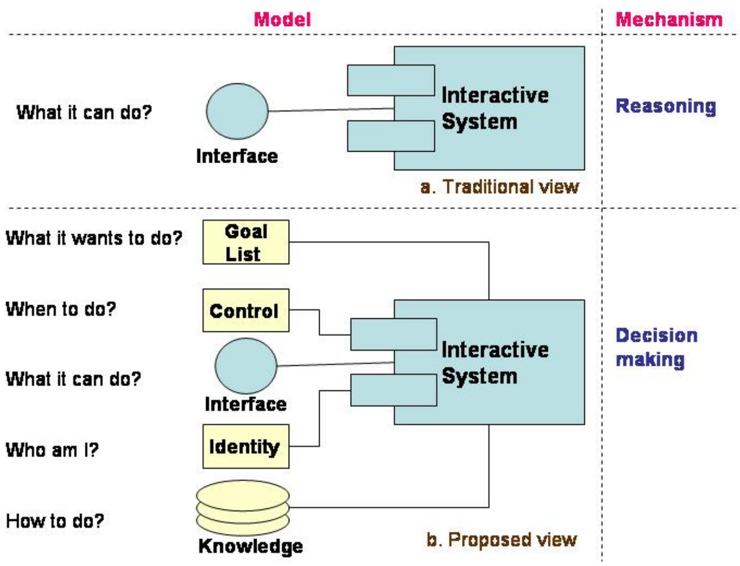

Figure 1: Enhanced Agent Interaction Model

has only one attribute, called interface (i.e., what they can do). Then the component can only be accessed through its interface (Figure 1a). In this case, the other agents requiring services of a certain software agent may consult directory and naming services (i.e., agent yellow pages) and use the services that the agent offers by adhering to the strict rules specified in the interface document for that agent. Experience shows that this limits the scope and applicability of the software agents, in the sense that autonomy, proactiveness may be compromised.

In this project, we propose an enhanced software agent model in which besides the interface (I), four more attributes are also specified: a goal list (G) (i.e., what the agent wants to do?); knowledge (K) (i.e., how to do?) that an agent can utilize to perform tasks autonomously; the thread of control (C) (when to do?) and identity (Id) (who to contact?) (Figure 1b). When interacting with the other agents, the I-G-K-C-Id attributes can be declared public, private or protected. Public means that the attribute is accessible and readable by all the other agents. Protected means that the attributes are visible only to a certain group of agents and private indicates that the attributes are not visible externally. Combination of the attributes and their states lead to various interaction scenarios.

Using this model the agent autonomy and proactiveness can be preserved if a proper decision making mechanism for the agents is devised and implemented. That is, the agent can decide upon the next task to accomplish using the current list of goals, interfaces, knowledge, identity and thread of control of self and the other agents with whom interacting.

Phase 3 implementing Objective 3:
Obviously, using the model in Figure 1b, each interaction scenario has certain properties and satisfying those properties requires implementation of certain reasoning and decision making mechanisms. In this phase, based on the above model, the MAS decision making catalog will be developed (**G31**).

Phase 4 implementing Objective 4:
A prototype decision making library to cover all the methods required for the interaction scenarios will be developed. A current version of the decision-making library, named Agent-DMSL, exists. It consists of Dempster-Shafer module, game theoretic certainty, risk and uncertainty modules. As for **G41** (Implementing the MAS decision making COTS library), the Agent-DMSL library will be enhanced to account for more modules, such as rule-based module, Bayesian belief networks (BBN), dynamic BBN and Ordered Weighted Averaging (OWA), etc. This COTS library will be used later to build business quality multi-agent systems (MAS) applications.

5. Anticipated Significance of the Work

This research shall have significant impacts for the economy, the software engineering discipline and for the society as a whole. Economical impact of the project is expected to be high. The factors that can make software systems useful assistant to human counterparts are the "quality of knowledge" and "improved cost-benefit ratio of decision-making" to make good use of knowledge. An important aspect of decision making is to get engaged in a competitive task to increase the profitability of self, company and society. A theory that can address this and a software system that can implement this feature efficiently will be highly appreciated in the areas such as marketing, setting price for a new product, setting target for production and particularly in electronic commerce providing services and/or selling goods online.

Lastly, the main impact in the discipline of software engineering will be in the reliability engineering and multi-agent systems (MAS) area. This research shows how to assess reliability of MAS and how to implement MAS using commercial-off-the shelf (COTS) components. This simplifies the MAS development and time-to-market and increases productivity.

References

[BIERMANN1998]: H.S. Bierman and L. Fernandez, "Game Theory with Economic Application," Addison-Wesley, 1998.

[CHIANG1984]: A.C. Chiang, "Fundamental Methods of Mathematical Economics," McGraw-Hill, 1984.

[FAR2003]: B.H. Far, "Methodological Support for Agent-Based System Interaction," Proceedings of the 1st International Conference on Agent Based Technologies and Systems ATS2003, pp. 27-40, 2003. 8. (Keynote Lecture)

[FRENCH1986]: S. French, "Decision Theory: An Introduction to The Mathematics of Rationality," Wiley, 1986.

[HARSANYI1968]: J.C. Harsanyi, "Games with Incomplete Information played by Bayesian Players," Management Science, vol. 14, no. 3, 5, 7, 1967 & 1968.

[IGLESAIS1997]: C. Iglesais, et al., "Analysis and Design of Multi-Agent Systems Using MAS-CommonKADS," in M.P. Singh et al. edts. Intelligent Agents IV: Agent Theories, Architecture and Languages, 1997.

[JENNINGS2000]: N.R. Jennings, "On agent-based software engineering," Artificial Intelligence, vol. 117, pp. 277–296, 2000.

[KAN2002]: S.H. Kan, "Metrics and Models in Software Quality Engineering," 2nd ed., 560 p., Addison-Wesley, 2002.

[KUHN1997]: H.W. Kuhn, "Classics in Game Theory," Princeton University Press, 1997.

[LIND2000]: J. Lind, "The MASSIVE Development Method for Multiagent Systems," In Proceedings of the 5th Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agents (PAAM2000), 2000.

[LYU1986]: M.R. Lyu (Edt.), "Handbook of Software Reliability Engineering," McGraw Hill, 1996.

[LIU2002]: B. Liu, "Theory and Practice of Uncertain Programming," Springer-Verlag, 2002.

[MUSA1998]: J.D. Musa, "Software Reliability Engineering," 391 p., McGraw-Hill, 1998.

[PARUNAK2002]: V. Parunak, S. Bruekner, J. Sauter, "ERIM Approach to Fine-Grained Agents," NASA/JPL Workshop on Radical Agent Concepts, 2002.

[PEARL1990]: J. Pearl, "Reasoning under uncertainty," Annual Review of Computer Science, 4:37-72, 1990.

[ODELL2001]: J.J. Odell, H.V. Parunak, B. Bauer, "Representing Agent Interaction Protocols in UML," Agent-Oriented Software Engineering, P. Ciancarini and M. Wooldridge eds., pp. 121-140, Springer, 2001.

[OSBORNE1994]: M.J. Osborne and A. Rubinstein, "A Course in Game Theory," MIT Press, 1994.

[RAO1995]: A.S. Rao, and M.P. Georgeff, "BDI agents: From theory to practice," In Proc. of the First Int. Conf. on Multi-Agent Systems (ICMAS-95), pp. 312-319, MIT Press, 1995.

[SHAFER1976]: G. Shafer, "A Mathematical Theory of Evidence," Princeton University Press, 1976.

[SHORTLIFFE1979]: E.H. Shortliffe, B.G. Buchanan and E.A. Feigenbaum, "Knowledge Engineering for Medical Decision Making: A Review of Computer-Based Clinical Decision Aids," Proceedings of IEEE, vol. 67, no. 9, pp. 1207-1224, 1979.

[SUBRAMAN2000]: V.S. Subramanian, et al., "Heterogeneous Agent Systems," MIT Press, 2000.

[YAGER1988]: R.R. Yager, "On Ordered Weighted Averaging Aggregation Operators in Multi-criteria Decision Making," IEEE Trans. SMC, no. 18, pp. 183-190, 1988.

[WEISS2000]: G. Weiss, Edt., "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence," 648 p., MIT Press, 2000.

[WOOLDRIDGE1998]: M. Wooldridge , N. R. Jennings, and D. Kinny, "A Methodology for Agent-Oriented Analysis and Design," In O. Etzioni, J. P. Muller, and J. Bradshaw, editors: Agents' 99: Proceedings of the Third International Conference on Autonomous Agents Seattle, WA, May 1998.