Using situation calculus for e-business agents

Conan C. Albrecht, Douglas D. Dean, James V. Hansen*

Marriott School of Management, Brigham Young University, Provo, UT 84602, USA

Abstract

As the Internet grows, it is becoming less feasible for customers and merchants to manually visit each web site, analyze the information there, and make sound business decisions regarding the trading of goods or services. To cope with this evolution, software agents can be designed that are capable of automating the more routine, tedious, and time-consuming tasks involved in current trading processes. At a higher level agents may also be able to negotiate and make autonomous decisions and commitments on behalf of their owners.

This paper describes an agent implementation using the situation calculus, which offers a possibly unifying paradigm for dynamic agents. Interesting applications are currently being developed. Our contribution is a situation calculus agent system developed for e-business. Ongoing work is focused on implementing this system in an open marketplace environment.
© 2003 Elsevier Science Ltd. All rights reserved.

Keywords: Internet; E-business; Software agent

1. Introduction

The Internet can be viewed as a large, distributed information resource, with connecting systems that are designed and implemented by many different organizations with various goals and agendas. Because of the breadth and flexibility of the Internet, e-business applications have grown rapidly in recent years, and this trend is expected to continue. Forrester Research and the Gartner Group estimate that e-business transactions will exceed $7 trillion by 2004. Thirty-seven percent of these transactions are estimated to be associated with electronic marketplaces, which provide resources to facilitate the interaction and exchange of commerce transactions among buyers, sellers, and other trading partners (Rahman & Bignall, 2001).

Today's e-business is increasingly characterized by the ability to create value through the gathering, synthesizing, and distribution of information. E-businesses progressively compete in real time rather than cycle time and operate through a constantly responsive dialogue with their customers and markets. Relationships among customers and markets are managed by technology through electronic channels. These technology-mediated channels are characterized by ongoing operations, which are subjected to measurement and tracking in new and innovative ways (Evans & Wurster, 2000).

Such applications are being facilitated by e-business architectures that provide opportunity for distributed, service-based applications, where service providers may not know about each other at design time. Services include such components as credit card charging, final fulfillment of orders, or publication of product information. These services are often distributed and run on different computers in geographically dispersed locations. Generally, services are created and offered by separate businesses, which specialize in a particular type of service. Such entities can often achieve economies of scale and provide levels of service beyond what a firm may be able to provide internally. This environment offers new opportunities for online business advantage.

Yet as the Internet grows in size and sophistication, it is becoming less feasible for customers and merchants to manually visit each web site, analyze the information there, and make sound business decisions regarding the trading of goods or services. It is inevitable that buyers may miss finding the best product or services in the vast milieu of information. In this new world, software agent technologies offer a robust paradigm for trading on the Internet and foreshadow a dramatically changed approach to conduct analysis, market research, and similar e-business functions. At one level, software agents can be designed to be capable of automating the more routine, tedious, and time-consuming tasks involved in current trading processes. At a higher level, agents can be designed to negotiate and make

* Corresponding author. Tel.: +1-801-422-2308; fax: +1-801-422-5933.
  E-mail address: james_hansen@byu.edu (J.V. Hansen).

autonomous decisions and commitments on behalf of their owners (Sandholm, 2001).

2. Agents and e-business

Implementation of a conventional strongly coupled system for electronic data interchange (EDI) between trading partners requires that a business know the providers and consumers of the various goods and services being traded so that transactions can be exercised between appropriate parties. In today's more open markets, determining this information in advance is extremely difficult or even impossible. Consider an electronic system to support open trading, where orders are made available to any qualified bidder. Requiring the system designer to determine the sender and recipient of each transaction in advance is not realistic. On the contrary, open trading applications are only loosely coupled, since not all of the necessary information is available when the system is designed.

Agents, however, can naturally support such applications. Instead of specifying the individual entities to be interconnected and their interfaces with one another, an agent-based system need identify only the classes of entities in the system and their impact on the environment. Because the agent is designed to interact with the environment rather than with specific other agents, it can interact appropriately with any other agent that modifies the environment within the range of variation with which other agents are designed to handle (Wooldridge, 2002).

Following Lesperance, Levesque, & Reiter (1999), we take an agent to be any active entity whose behavior is usefully described through mental notions such as knowledge, goals, abilities, commitments, and the like. At a low level, agents are currently being used to help suppliers reduce inventory carrying costs and marketing costs by offering special discounts unavailable to the general public. At a higher level agents are being programmed with rules or learning capability that have led to their being termed 'intelligent' agents.

Intelligent agents are being used to help businesses determine which customers are most likely to be interested in purchasing the discounted items, to compare product and service information, to help users narrow the number of sites that contain desired information, and negotiating business transactions: prices, delivery schedules, quality, warranties, etc. Firms are also using intelligent agent technology to continuously conduct environment scans. For example, an attorney that specializes in patent law can use pull technology to continuously monitor multiple sites and signal any developments such as new patent-related court cases or court decisions. (Greenstein & Vasarhelyi, 2002)

Central to successful agent implementation is the issue of how to program agents to perform their functions. In one sense, agent programming can be viewed as a generalization of object-oriented programming; yet the notion of an agent is more complex than that of an object. Hence, it is important that tools for modeling and designing agents be based on sound theoretical foundations. Many of today's agent paradigms are ad hoc or otherwise difficult to evaluate (Reiter, 2001).

Although several powerful formalisms exist, finding the right formalism is a nontrivial challenge, as it must provide a level of expressiveness that serves the practical problems at hand in a tractable way (Lesperance, Levesque, Lin, & Scherl, 2000). Despite the range of choices, there is general agreement that formal methods do help in the long run in helping to develop a clearer understanding of problems and solutions (Kraus, 2001).

This paper describes an approach and application grounded in theoretical and practical developments arising from recent research in cognitive robots and software agents. In particular the development of extensions to situation calculus opens new possibilities for agent programming (Boutelier, Reiter, Soutchanski, & Thrun, 2000).

Situation calculus allows any defined set of complex action expressions to be viewed as the programming language for agents. That language can then be used to model the behavior of a set of agents and to actually implement them (Reiter, 2001).

Relevant research in artificial intelligence aims at explaining and modeling intelligent behavior in terms of computational processes. The classical approach assumes intelligent behavior to be a result of correct reasoning on correct representations. In turn, reasoning is understood by means of formal logic. In research areas such as cognitive robotics and software agents, this approach to AI is applied to a crucial aspect of intelligent behavior, that of acting in a dynamic world. Situation calculus addresses this issue formally and directly (Lesperance et al., 2000).

3. High-level agent programming using situation calculus

The construction of autonomous agents is paramount in artificial intelligence, with considerable research devoted to methods that will ease the burden of designing controllers for such agents. There are two principal ways in which the complexity of devising controllers can be constructed. The first is to provide languages, with which a programmer can specify a control program with relative ease, using high-level actions as primitives, and expressing the necessary operations in a natural way. The second is to simply specify goals and provide the agent with the ability to plan appropriate courses of action that achieve those goals. In this way the need for explicit programming is avoided (Wooldridge, 2002).

The development of this type of computational agent has been the subject of considerable recent work. The core