ELSEVIER

Decision Support Systems 27 (2000) 355–366

Decision Support Systems

www.elsevier.com/locate/dsw

Interface agents: caveat mercator in electronic commerce

Daniel G. Conway [1], Gary J. Koehler *

Decision and Information Sciences, Warrington College of Business, Stuzin 351, University of Florida, P.O. Box 117169, Gainesville, FL 32611-7169, USA

Accepted 18 August 1999

Abstract

Electronic commerce has opened new opportunities for buyers and sellers. Consumers can do things in an on-line environment that are simply not possible in face-to-face transactions. In this paper, we push this observation by examining and using a new type of software agent to convert merchant interfaces into middleware thus enabling one to assess and optimize their interactions with all the computing support available in today's Decision Support System (DSS) environments.

Keywords: Electronic commerce; Software agents; Internet gaming; Agent blocking; Software interfaces

1. Introduction

Electronic commerce has opened new opportunities for buyers and sellers. Buyers have on-line search capabilities for finding products and services, for comparing prices and for ease of purchase. Sellers can reach and service more customers at lower costs and provide on-line support. However, anyone can easily start an electronic enterprise — both legitimate companies and others. As a result, many consumers have concerns about vendor viability, trust-worthiness of the companies and quality of products or services. Most already worry about secure transactions. As a result, many buyers have taken a *caveat emptor* — buyer beware — approach to electronic commerce.

We explore the converse issue — the issue of *caveat mercator* — seller beware. Consumers can do things in an on-line environment that are simply not possible otherwise. In an on-line environment, consumers have computing resources not normally available in face-to-face transactions. Financial investors have long known this — computers can monitor prices across markets and find arbitrage opportunities that would be impossible to spot, to then compute optimal responses and to execute actions within the brief windows of opportunity available.

A more pertinent example of caveat mercator is Andersen Consulting's BargainFinder agent [2]. BargainFinder searched a number of CD outlets on the Internet locating the best price of a desired item. Eventually a number of these vendors realized their profitability was dropping and took measures to block BargainFinder's interactions. This is a simple example of caveat mercator. The emerging Internet environment opened buyer opportunities not faced by

* Corresponding author. Tel.: +1-352-846-2090; fax: +1-352-392-5438; e-mail: koehler@ufl.edu

[1] Tel.: +1-352-392-9599; e-mail: conwaydg@ufl.edu.

0167-9236/00/$ - see front matter
PII: S0167-9236(99)00046-9

real-world merchants. These opportunities resulted from inexpensive searching and comparison capabilities.

Although BargainFinder illustrates a caveat mercator situation, we wish to explore a more complex setting where computing and decision making resources are also employed. These tools enable an agent to discover hard-to-see opportunities — or opportunities never exploited when the tools could not be used or not used effectively. These may not be anticipated by merchants nor easily thwarted. BargainFinder was blocked by merchants who stood to lose. This blocking was easy since BargainFinder worked from a central site. Jango [16] circumvented this by working from user sites.

Typically software agents are long-lived, semi-autonomous, proactive, and adaptive. An ideal consumer agent would interact with a merchant's agent to find goods, negotiate price, etc. These agents would have to learn and adapt to changing conditions while still striving towards the overall goals of their owner (see Ref. [14] for a good survey of agent usage in electronic commerce).

Unfortunately, most on-line consumer-merchant interactions still require direct manipulation of an interface. We take this as a given and propose a software agent that knows how to interact with interfaces. This reduces any merchant application to middleware that can be manipulated in the same manner a human would manipulate it. More importantly, our agent can be endowed with Decision Support System (DSS) and AI capabilities in ways never anticipated by merchants.

This research proposes a software agent that works directly through a merchant's web-site interface. This agent treats an e-commerce site as middleware and responds with mouse clicks, typed text, etc. just as a human might. We call this an interface agent. This behavior hides the software agent's activities making it appear as if a human was dealing with the application. However, the software agent can easily bring to bear sophisticated information processing and decision-making tools.

We propose that our interface agent can exploit opportunities not present in face-to-face transactions. To test this hypothesis, we developed a prototype and used it in hundreds of thousands of transactions to exploit an opportunity not practical in similar, direct human interactions. Indeed, the very number of transactions undertaken would not have been possible by direct human interactions.

This study shows that an interface agent can seriously compromise a merchant's business. Furthermore, we explore measures merchants might take to protect themselves. However, none of our proposals is fool proof. Our results indicate that caveat mercator is a serious consideration. Companies should understand this threat before launching an e-commerce business. This paper brings to light both new DSS opportunities (made possible by an interface agent) and exposes the need for businesses to understand and counter this IT-based threat.

In Section 2 we briefly review agent literature and propose our interface agent. In Section 3 we discuss our test environment, give relevant background information and describe our specific implementation. We tested this model in a real-world setting. Following a long tradition in Artificial Intelligence research, we focused on a gaming environment. The setting we chose to study was Internet Gaming. Besides being fun, it presented an environment that exhibited all the attributes we sought. In Section 4 we show the results of the usage of our software agent. Section 5 presents methods that merchants can use to block traditional unwanted agents. It is important to understand these tactics to see what advantages an interface agent provides. Surprisingly, these and many common security methods are impotent against interface agents. We present some possible steps merchants might employ against interface agents. Finally, in Section 6 we present our conclusion and directions for future research.

2. Software agents

An intelligent agent is a software entity that possesses some type of intelligence and performs autonomous operations for a human (see Brenner et al. [4] for example). Agents must interact with an environment. This necessitates communication, cooperation and coordination. An agent is evaluated on its capability to learn, to goal-seek, to react to the environment, to provide autonomous behavior, and to move about a networked environment.